



**Calhoun: The NPS Institutional Archive**  
**DSpace Repository**

---

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

---

1975

# Digital simulation as an approach to ESM software development.

Stanga, Robert Daniel

Monterey, California. Naval Postgraduate School

---

<http://hdl.handle.net/10945/20823>

---

*Downloaded from NPS Archive: Calhoun*



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

<http://www.nps.edu/library>

DIGITAL SIMULATION AS AN APPROACH  
TO ESM SOFTWARE DEVELOPMENT

Robert Daniel Stanga

OWLEY KNOX LIBRARY  
N/LL POSTGRADUATE SCHOOL  
MONTEREY, CALIFORNIA 93940

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



# THESIS

DIGITAL SIMULATION AS AN APPROACH  
TO ESM SOFTWARE DEVELOPMENT

by

Robert Daniel Stanga, Jr.

March 1975

Thesis Advisor:

S. Jauregui

Approved for public release; distribution unlimited.

T170811



| REPORT DOCUMENTATION PAGE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                       | READ INSTRUCTIONS<br>BEFORE COMPLETING FORM                          |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|----------------------------------------------------------------------|
| 1. REPORT NUMBER                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER                                        |
| 4. TITLE (and Subtitle)<br>Digital Simulation as an Approach<br>to ESM Software Development                                                                                                                                                                                                                                                                                                                                                                                                                               |                       | 5. TYPE OF REPORT & PERIOD COVERED<br>Master's Thesis;<br>March 1975 |
| 7. AUTHOR(s)<br>Robert Daniel Stanga, Jr.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                       | 6. PERFORMING ORG. REPORT NUMBER                                     |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Naval Postgraduate School<br>Monterey, California 93940                                                                                                                                                                                                                                                                                                                                                                                                                    |                       | 8. CONTRACT OR GRANT NUMBER(s)                                       |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Naval Postgraduate School<br>Monterey, California 93940                                                                                                                                                                                                                                                                                                                                                                                                                        |                       | 10. PROGRAM ELEMENT, PROJECT, TASK<br>AREA & WORK UNIT NUMBERS       |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)                                                                                                                                                                                                                                                                                                                                                                                                                                               |                       | 12. REPORT DATE<br>March 1975                                        |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                       | 13. NUMBER OF PAGES<br>96                                            |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                       | 15. SECURITY CLASS. (of this report)<br>Unclassified                 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                       | 15a. DECLASSIFICATION/DOWNGRADING<br>SCHEDULE                        |
| 16. DISTRIBUTION STATEMENT (of this Report)<br><br>Approved for public release; distribution unlimited.                                                                                                                                                                                                                                                                                                                                                                                                                   |                       |                                                                      |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)                                                                                                                                                                                                                                                                                                                                                                                                                                |                       |                                                                      |
| 18. SUPPLEMENTARY NOTES                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                       |                                                                      |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number)                                                                                                                                                                                                                                                                                                                                                                                                                                        |                       |                                                                      |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number)<br><br>Selective problem areas relating to computer based ESM system software development are discussed. Digital simulation is presented as a practical approach to solving software development problems. An algorithm to simulate a hard-wired EW signal preprocessor is developed. An application oriented high level language version of the algorithm is presented.<br>A FORTRAN program which generates realistic data for signal |                       |                                                                      |



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

analysis is described. Using the simulated data, an instruction level simulation program, and a language compiler, the algorithm is tested using an IBM 360 computer.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)



Digital Simulation as an Approach  
to ESM Software Development

by

Robert Daniel Stanga, Jr.  
Lieutenant, United States Navy  
B.S.A.E., Auburn University, 1968

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the  
NAVAL POSTGRADUATE SCHOOL  
March 1975



## ABSTRACT

Selective problem areas relating to computer based ESM system software development are discussed. Digital simulation is presented as a practical approach to solving software development problems. An algorithm to simulate a hard-wired EW signal preprocessor is developed. An application oriented high level language version of the algorithm is presented.

A FORTRAN program which generates realistic data for signal analysis is described. Using the simulated data, an instruction level simulation program, and a language compiler, the algorithm is tested using an IBM 360 computer.



## TABLE OF CONTENTS

|             |                                                                                                                                                                       |    |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| I.          | INTRODUCTION -----                                                                                                                                                    | 8  |
| A.          | STATEMENT OF THE PROBLEM -----                                                                                                                                        | 8  |
|             | 1. ESM Hardware/System Description -----                                                                                                                              | 9  |
|             | 2. Man-Machine Interface -----                                                                                                                                        | 11 |
|             | 3. ESM Software -----                                                                                                                                                 | 12 |
| B.          | SCOPE OF THE THESIS -----                                                                                                                                             | 12 |
| II.         | COMPUTER SYSTEM FOR ESM SOFTWARE DEVELOPMENT -                                                                                                                        | 15 |
| A.          | GENERAL DESCRIPTION -----                                                                                                                                             | 15 |
| B.          | APSS -----                                                                                                                                                            | 16 |
| C.          | PL/ATAC -----                                                                                                                                                         | 17 |
| III.        | ESM ENVIRONMENT/IFM SIMULATION -----                                                                                                                                  | 19 |
| IV.         | THE PULSE TRAIN SEPARATOR -----                                                                                                                                       | 27 |
| A.          | CONCEPT OF OPERATION -----                                                                                                                                            | 27 |
| B.          | PTS SIMULATION PROGRAM -----                                                                                                                                          | 28 |
| V.          | DISCUSSION OF SIMULATION RESULTS -----                                                                                                                                | 34 |
| A.          | SUMMARY OF RESULTS -----                                                                                                                                              | 34 |
| B.          | CONCLUSIONS -----                                                                                                                                                     | 35 |
| C.          | RECOMMENDATIONS -----                                                                                                                                                 | 36 |
| APPENDIX A. | Instruction for installation and use<br>of the ATAC Programming Support System<br>(APSS) on the IBM 360/67 operating system<br>at the Naval Postgraduate School ----- | 37 |
| APPENDIX B. | Data Simulation Program<br>FORTRAN Listing and Output -----                                                                                                           | 43 |
| APPENDIX C. | Pulse Train Separator Program<br>PL/ATAC Listing -----                                                                                                                | 52 |



|                                                 |    |
|-------------------------------------------------|----|
| APPENDIX D. Pulse Train Separator Program ----- | 68 |
| ASSEMBLY LANGUAGE LISTING AND OUTPUT -----      | 91 |
| BIBLIOGRAPHY -----                              | 94 |
| INITIAL DISTRIBUTION LIST -----                 | 95 |



### ACKNOWLEDGEMENT

I am deeply grateful to Applied Technology Division, Itek Corporation, Sunnyvale, California for the use of their facilities and the generous assistance and experience of their personnel, particularly the ATAC Support Software Group. I am particularly indebted to my wife, Phyllis, whose faith and devotion made this work possible and to whom it is dedicated.



## I. INTRODUCTION

### A. STATEMENT OF THE PROBLEM

Electronic Warfare Support Measures (ESM) is that division of Electronic Warfare (EW) concerned with exploitation of enemy use of the electromagnetic spectrum. ESM includes actions taken to search for, intercept, locate, and analyze electromagnetic radiations to support tactical military operations.

The continuously increasing density and complexity of today's electromagnetic environment, created by friendly as well as hostile forces, has fostered the requirement for more effective performance and greater sophistication in operational ESM systems. Current receiver technology has kept pace with this escalation, and signal intercept is virtually assured. However, the resulting increased data rates, existing and forecast manpower limitations, and limited reaction time available to the operational commander have imposed a requirement for faster and more accurate signal analysis, classification, and interpretation. Currently, major efforts are being put forth to achieve an effective ESM system design incorporating high speed digital computers for the purposes of automatic signal analysis and system control.

To perform effectively in present and future environments, an automated ESM system should exhibit the following characteristics:



1. Broad frequency range.
2. Simultaneous coverage of entire frequency range.
3. High probability of intercept.
4. Low false alarm rate.
5. Capability to recognize pulse or radar-like signals.
6. Capability to determine parameters associated with radar pulse trains.
7. Capability to classify signals according to type and source.
8. Direction finding capability.
9. Capability to handle data rates.
10. Capability to detect signals at ranges to radar horizon.

The successful design and development of a system exhibiting these characteristics rest upon three equally important elements: system hardware, system software, and the man-machine interface.

#### 1. ESM Hardware/System Description

A discussion of ESM hardware as a specific problem area is beyond the scope of the thesis. A system concept compatible with the state of the art approach is presented here as a basic foundation for this study and future efforts in the ESM arena.

Conceptually, an airborne ESM system with the characteristics previously described could be implemented utilizing a combination of Instantaneous Frequency Measurement (IFM) and Superhet receivers. The IFM receivers, operating in conjunction with a high-gain pencil beam antenna, would cover the frequency bands of interest and function as the primary acquisition receivers of the system. In addition to contributing to increased system sensitivity, the high-gain pencil



beam antenna functions to reduce the instantaneous data rate and, in conjunction with azimuth centroiding, provides an accurate means of determining signal direction of arrival.

Additional data rate reduction is accomplished by routing each IFM output through a Pulse Train Separator (PTS), which functions as a signal preprocessor to separate stable pulse trains from the received data. Parameter measurements are performed on those stable trains separated and a computer word is generated to tag the associated emitter. The required capacity of the PTS in number of pulse trains separated is considered paramount to system operational effectiveness.

A computer word is generated for each pulse associated with non-stable pulse trains and those stable pulse trains exceeding the capacity of the PTS. These data are routed through a residual channel of the PTS to a high-speed digital computer for deinterleaving and parameter measurement. The parameters associated with each signal are then passed to a main frame computer for classification, threat analysis, and tactical environment update.

The IFM system employs an omni antenna for side lobe suppression to prevent signal poke-through in the receiving antenna minor lobes. The IFM sensitivity and antenna gain provide a capability for radar detection out to the radar horizon for aircraft altitudes in the neighborhood of 30,000 feet.

Priority signals, unclassifiable signals, and signals too complex for automatic analysis are automatically assigned



to digitally controlled superhet receivers for refined analysis. The superhet receivers cover the frequency bands of interest and operate in conjunction with an omni or low-gain broad beam antenna.

## 2. Man-Machine Interface

The human operator element of an automated ESM system, though normally an area addressed by human factors engineers, impacts heavily on the software requirements and will be discussed here in that regard.

Automation does not eliminate the requirement for human operators, but it does significantly affect the role the operator plays in the ESM system and it may seriously retard the operator's proficiency and ability to function efficiently as a back up in the event of a system failure. Thus, in deciding which system functions should be automated and which should be assigned to the human operator, it is imperative that the operator be assigned those functions that maximize the use of his capability to exercise judgment, improvise, reason inductively, and profit from experience.

The human operator must be able to control and direct the operation of the system. To do so, he must be provided relevant information by the system at the proper time, in the proper format, and on the proper displays. To obtain efficient and reliable software, these requirements must be clearly and accurately defined in the planning stage of the ESM system development.



### 3. ESM Software

It is felt that software represents the most troublesome problem in the design and development of an automatic ESM system. Assuming cost and risk vary directly with complexity, ESM software will account for a substantial percentage of total system costs and be the riskiest part of the system development. Since system performance hinges on software performance, implementation of an effective software test plan would serve a threefold purpose of cost reduction, risk reduction, and increased system performance.

The ability to successfully implement such a test plan requires a detailed description of the systems functions and a software modular building process. Each software module or set of modules is then integrated into the system and tested against sets or subsets of system functions according to prescribed test plans.

#### B. SCOPE OF THE THESIS

Simulation, via computer software, is one technique that is useful in software development and verification or testing. Realistic data, created by successful simulation of the ESM system hardware operating in a dynamic environment, utilized as inputs to a high-speed digital computer would provide the framework for implementation, testing, and optimization of ESM signal processing algorithms. The inherent ability to control the density of the electromagnetic environment through simulation would provide the capability to analyze worst case



situations, thereby enabling system designers to estimate computer capacity and reserve capacity requirements as well as data rate reduction requirements.

Given these simulation capabilities, an instruction-level simulation of the target computer provides a powerful means of debugging software and algorithm testing for accuracy and speed of operation. Applied Technology Division of Itek Corporation, Sunnyvale, California, has provided the Naval Postgraduate School with the programming support system for their ATAC computer. Instruction level simulation is only a part of this software package which includes the compiler programs for Programming Language for ATAC (PL/ATAC), a high level language developed for ATAC applications programming. Acquisition of the ATAC Programming Support System (APSS) gives impetus to the utilization of simulation as a technique for ESM software development and testing at the Naval Postgraduate School.

Total hardware simulation of the ESM system previously described is beyond the scope of the thesis. Total system simulation does, however, lend itself to a software "build" process and as a first effort of the process, three specific objectives of this work are as follows:

1. Provide a digital simulation of an IFM receiver operating in conjunction with a high-gain, pencil beam DF antenna in a dynamic radar environment.
2. Provide a digital simulation of a hard-wired pulse train deinterleaver fed by an IFM receiver.
3. Describe and demonstrate the capabilities of APSS, particularly, the PL/ATAC programming language.



The overall objective of this work is to provide a building block as a basis for follow on study of ESM system software at the Naval Postgraduate School.



## II. COMPUTER SYSTEM FOR ESM SOFTWARE DEVELOPMENT

### A. GENERAL DESCRIPTION

APSS was written for Applied Technology's Airborne Computer (ATAC). ATAC, specifically designed for Electronic Warfare applications, is well suited for ESM software design, development and testing. A number of features of this 16-bit, fixed point, two's complement computer permit the establishment of a meaningful baseline for determining the relative merit of algorithms applicable to computer based ESM systems. Special features of interest are:

1. 437 nanosecond major cycle time.
2. Sixteen general registers which may be used as accumulators or as index registers.
3. Microprogrammable processor, permitting optimization of the instruction set for specific applications.
4. Memory types can be mixed in any combination of low power Read Only Memory (ROM), fast bipolar Random Access Memory (RAM), and slower core memory devices.
5. Memory expandable to 65K words.
- 3 6. Four basic instruction addressing modes: register, immediate, direct, and direct indexed.
7. Three supplemental addressing modes: register indexed, immediate short, and direct register.
- 77 8. Provisions for up to 16 Direct Memory Access (DMA) channels.
- 4 9. Provisions for up to four simultaneously active DMA channels.
- 6 10. Twelve hardware priority interrupts.
- 8 11. Modular, compact construction.

Additional detailed information, including the instruction set, may be obtained from ATAC, Applied Technology Airborne Computer, Vol. I, Principles of Operation.



## B. ATAC PROGRAMMING SUPPORT SYSTEM (APSS)

This software support system allows for the preparation and debugging of programs via an IBM 360/370 computer system. APSS, through simulation of the ATAC computer and associated peripheral devices, permits the applications programmer to optimize programs for execution speed and memory utilization. The APSS system is written in a combination of PL/1, FORTRAN IV, and IBM assembler language and includes the programs listed below.

- APSS Monitor
- ATAC Instruction Assembler
- ATAC Instruction-Level Simulator
- Relocating and Linking Loader
- Microcode Assembler
- Microcode-Level Simulator
- PL/ATAC Compiler
- Library Control System
- Subroutine Library

Arrangements were made with Applied Technology to use portions of APSS and consequently, the ATAC Instruction-Level Simulation Program and the PL/ATAC Compiler are currently operational on the IBM 360/67 at the Naval Postgraduate School. Information concerning the implementation of these programs and instructions pertaining to the use of the programs are contained in Appendix A. Additional APSS information is available in ATAC, Applied Technology Airborne Computer, Volume 2, ATAC Programming Support System (APSS). The documents in this volume provide a detailed description of how to set up and use the APSS system. Those documents applicable to this work are:



1. Simulation of instruction execution.
2. Tracing option of the 16 general registers. The contents of each register can be printed after the execution of an instruction.
3. Accumulated program execution time printed with each line of trace.
4. Instruction execution histograms by frequency of execution or time of execution can be created and printed.
5. Partial and total simulated memory dump options.
6. Initialization of program data arrays allowing data to be read into virtual program memory prior to program simulation.

The APSS Instruction-Level simulator program includes a large repertoire of error messages to assist the applications programmer in debugging job management and ATAC instruction usage errors.

#### C. PL/ATAC

PL/ATAC, Programming Language for ATAC, is a high level, block structured language designed specifically for the ATAC computer. The programming structure and numerous capabilities of the language are demonstrated in Appendix C via the Pulse Train Separator simulation program written in PL/ATAC. In addition to hardware simulation, a specific objective in writing this program was to provide programming examples of the PL/ATAC language to assist the interested programmer in learning PL/ATAC. The PL/ATAC manual (Volume 2, Section II, ATAC) is an excellent reference but may present some difficulty if used as a learning text.

PL/ATAC is a powerful language providing the programmer with programming control approaching that of assembly language



without compromising the high level of the compiler language. Additionally, ATAC assembly language may be intermixed with PL/ATAC increasing programming flexibility and control with no serious degradation of program readability.

An analysis to determine the effectiveness of applications programming in the PL/ATAC language versus ATAC assembly language was conducted by the author utilizing an assembler optimized ATAC benchmark program. The results of this analysis indicated a memory utilization increase by a factor of approximately 2.3 times the ATAC assembly language requirement and an execution time increase by approximately 2.4 times the ATAC assembly language execution time.

The PL/ATAC programmer loses some flexibility, as compared to the assembly language programmer, due to the limitation in the number of registers available (a maximum of five) to him. Also, the PL/ATAC compiler currently installed at the Naval Postgraduate School does not have a double precision capability. Double precision calculations, if required, must be accomplished in assembly language resulting in increased core space and execution time required.



### III. ESM ENVIRONMENT/IFM SIMULATION

Investigation revealed the existence of a computer program [Ref. 1] designed to simulate an IFM receiver operating in conjunction with an omni antenna in a dynamic radar environment. This program was modified to incorporate a rotating DF antenna system. Routines were added to generate unstable pulse trains in the form of pulse 'jitter' and/or 'stagger'. Documentation was added to facilitate program readability.

The DATA Simulation Program generates simulated pulsed radar signal level data arriving at an IFM receiver via a rotating antenna system. Current capacity allows for simulation of up to ten emitter platforms with up to five emitters per platform. This capacity may be increased by increasing the array dimensions of the parameters associated with each emitter-platform combination. Emitter platform and IFM platform initial position information in X-Y grid coordinates, course in degrees clockwise from North, and speed in knots are set by the program input data.

Emitter characteristics, set by the program input data, consist of transmitter frequency in MHz, pulse repetition interval (PRI) in milliseconds, pulse width (PW) in microseconds, transmitter power in kilowatts, antenna gain in db, antenna beamwidth in degrees, antenna scan rate in seconds per revolution, and pulse jitter and/or stagger characteristics. A  $\sin(x)/x$  antenna radiation pattern is assumed for all emitters.



IFM receiver characteristics, set by the program input data, consist of receiver threshold sensitivity in dbm, upper and lower frequency limits of the passband in MHz, minimum time measurement capability in microseconds, antenna gain in db, antenna scan rate in seconds per revolution, and antenna beamwidth in degrees.

Additional program inputs include simulation start and stop times in seconds, frequency bin size of the IFM digital processor, and random number generator calling arguments. Complete input data description is depicted in the program listing following the description of the program variables. A complete program listing is contained in Appendix B.

Utilizing the input data described, the Data Simulation Program computes the position, course, speed, and range of each emitter platform relative to the IFM platform. The closest point of approach (CPA) and distance to the CPA is computed for each emitter platform to determine if and when a platform will be within the range of detection of the IFM. Those emitters out of the detection range during the simulation are listed as such. The IFM antenna's and all emitter antennas' initial pointing directions are set by a random process utilizing a uniform random number generator. The time at which each emitter platform enters the capture area of the IFM antenna's main beam is calculated and used as a basis for computing a random transmit time for the emitters associated with that platform.



Emitter maximum effective radiated power ( $P_E$ ) is then computed utilizing the mathematical model given by

$$P_E = P_t G_t \lambda^2 / 16\pi^2$$

where

$P_t$  = Transmitted Power,

$G_t$  = Emitter Antenna Gain,

and  $\lambda$  = Wavelength of Signals.

Emitter maximum range of detection ( $R_m$ ) is computed by:

$$R_m = (P_E G_r / T_h)^{1/2}$$

where

$G_r$  = IFM Antenna Gain,

and  $T_h$  = IFM Receiver Threshold Sensitivity.

For each emitter within detection range, the minimum antenna gain ( $S_{min}$ ) required to break the IFM threshold sensitivity is then computed by

$$S_{min} = T_h / G_r P_r r^2$$

where

$r$  = Distance to the Emitter.



Using the value of emitter transmit time previously calculated, the emitter antenna pointing direction is calculated. Emitter antenna gain as a function of the relative angle between the emitter antenna and the IFM antenna is then calculated and compared with the minimum antenna gain required to break the IFM threshold sensitivity. If the signal level is sufficient to break the IFM threshold, the time of arrival (TOA) of the pulse at the IFM receiver is calculated.

If the pulse does not break the IFM threshold, the emitter transmit time is incremented by the emitter PRI, the antenna position is updated, and the antenna gain calculation process is repeated until a signal level of sufficient strength is transmitted. Should no pulse be received in the scan time of one beamwidth of the IFM antenna, the subsequent scan of the receiver antenna is checked and the process continues until a pulse is received or until simulation stop time.

The time of arrival of the first pulse from each emitter is stored in an array and, upon completion of this process for all emitters, the array is searched for the lowest time of arrival. The receiver antenna bearing at this TOA is calculated and provided as output data along with the parameters of the associated emitter. The emitter platform position is updated and the time of arrival of the next pulse from this emitter is calculated and stored in the array. The array is then searched for the next lowest time of arrival and the process repeated until simulation stop time.



The output of the Data Simulation Program consists of parameters associated with each pulse received by the IFM receiver in the proper time sequence. These parameters include the frequency, pulse width, bearing and time of arrival.

This data, as prepared for the ATAC simulation programs, consists of five integers for each pulse. These integers represent the following:

1. Emitter bearing. An integer in the range of 0-359. This word represents the IFM antenna bearing at the time of arrival of the pulse.
2. Frequency bin. The bins are numbered in ascending order of frequency from 1-128. This range may be altered as desired by the programmer.
3. Pulse width. The pulse width word is an integer in the range 0-65,535. The time scaling factor is arbitrary. Current convention is that the least significant bit of the pulse width word corresponds to .1 microseconds. This allows for representation of pulse widths from 0-6.5 milliseconds to within 0.05 microseconds.
4. Time of arrival. Time of arrival is expressed as a double precision (32 bit) integer. Word 4 is the least significant half of this integer and holds time values up to 0.84 seconds with the least significant bit corresponding to 25.6 microseconds.



5. Time of arrival. The upper half of the time word holds time values up to 15.3 hours.

The Data Simulation Program produces these outputs in printed format and in a punched card format suitable for acceptance by the ATAC simulation program. A sample of the program output is contained in Appendix B following the program listing. The program flow chart is illustrated in Figures 1 and 1a.



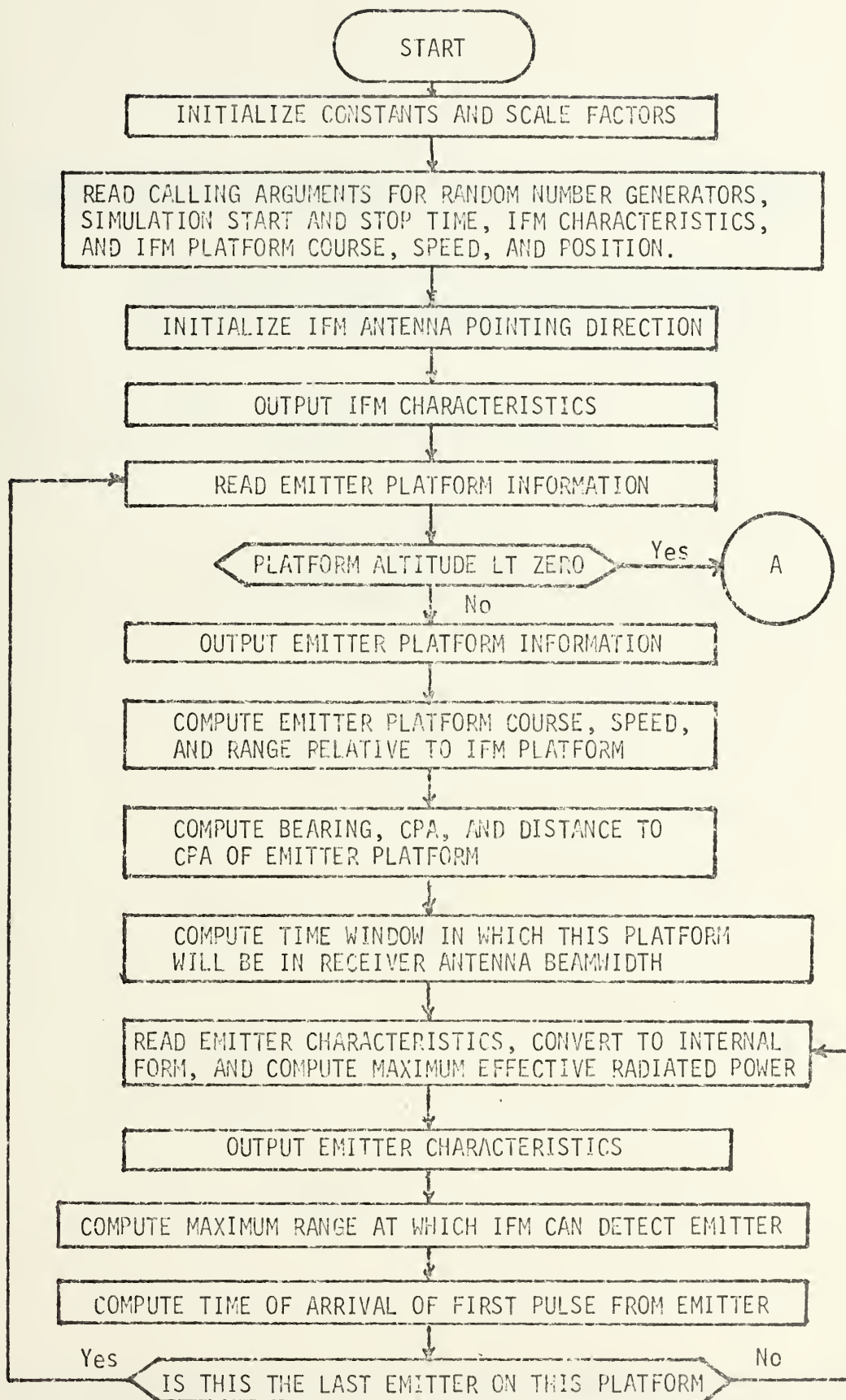


FIGURE 1. Data Simulation Program (Simplified Flowchart)



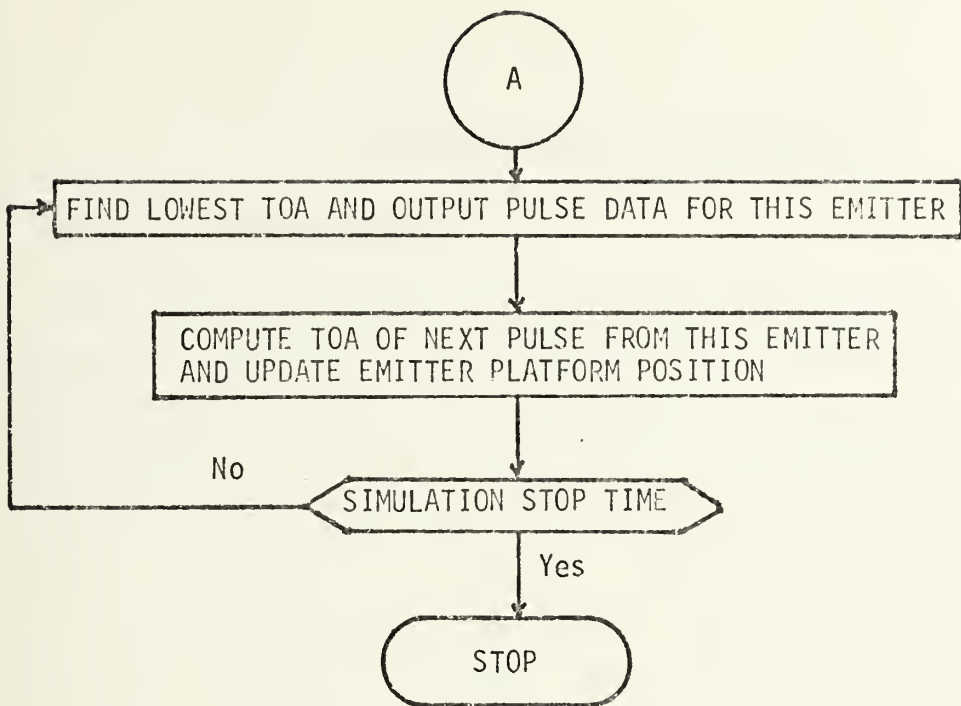


FIGURE 1a. Data Simulation Program (Simplified Flowchart)



#### IV. THE PULSE TRAIN SEPARATOR

##### A. CONCEPT OF OPERATION

Data received by the IFM are routed to the Pulse Train Separator (PTS). If a pulse train has a stable pulse repetition frequency (PRF) the PTS locks on the train and tracks the train for a period of time determined by the receiver antenna scan rate and beamwidth. The signal's pulse width (PS) and pulse repetition interval (PRI) are measured and digitized along with azimuth, time first seen and frequency. These five parameters are used to tag the associated emitter and are routed to the main frame computer for emitter classification. Direction of arrival is computed by azimuth centroiding based on the receiver antenna bearing at the time of arrival of the first and last pulse of the signal. The possibility exists that, due to  $\sin(x)/x$  characteristics of emitter antennas, centroiding will be performed on ragged signal inputs. This is not considered degrading to system performance as meaningful direction of arrival information will still result.

An inherent characteristic of the Pulse Train Separator is that those stable trains with the lower values of PRI will be separated first. This feature of the PTS is also desirable as the data rate reduction role of the device is enhanced. The required capacity of the device in this role of data rate reduction has not been established and is considered a most



important factor in the overall ESM system design. Failure to provide the proper level of preprocessing could result in system software and hardware overloading and, consequently, degraded system performance.

Those stable signals exceeding the capacity of the Pulse Train Separator and non-stable signals such as frequency agile signals, jittered pulse trains, staggered pulse trains, and other exotic signals are routed through a residual signal channel of the pulse train separator to the high speed mini computer for parameter measurement. For each of these pulses received, a computer word will be generated for frequency, time of arrival, pulse width, and direction of arrival. Pulse deinterleaving and such parameter measurements as pulse repetition interval and pulse jitter and stagger characteristics will be performed by the mini computer for emitter classification.

#### B. PTS SIMULATION PROGRAM

The Pulse Train Separator Simulation Program is designed to simulate a hardware preprocessing system with a capability of separating up to a maximum of ten stable pulse trains from the output of an IFM receiver.

Input data to the program consist of a series of five integer words for each pulse generated by the Data Simulation Program; direction of arrival in degrees clockwise from North, frequency by bin number, double precision time of arrival (two words) in seconds with the least significant bit



corresponding to 25.6 microseconds and pulse width in seconds with the least significant bit corresponding to .1 microseconds. These data are loaded into a large input buffer prior to the start of the program simulation.

Each pulse within the buffer is compared in frequency with previously received pulses within the same receiver antenna beamwidth. If no match in frequency occurs the pulse is considered a new signal and a pointer address is stored in a New Signal Pointer File. If a frequency match occurs, the pulse width of the signals are compared. Upon obtaining a match in pulse width the matched pulse's pointer address is stored in a five element array within the New Signal Pointer File. Upon receipt of five pulses of a pulse train, the program calls a local re-entrant procedure (PRICOMP) wherein the stability of the pulse train is determined. For purposes of this simulation a pulse train is considered unstable if the variation in time interval between pulses is greater than 51.2 microseconds and/or three or more missing pulses are encountered in ragged signals. Upon determining a pulse train to be stable, the pulse repetition interval of the signal is calculated and the signal is flagged as a stable train. The number of available channels in the Pulse Train Separator is then reduced by one and the signal is zeroed in the input buffer as each additional pulse is received. Parameters associated with this pulse train are stored and the signal bearing is updated as each pulse is received through incrementation of the input buffer.



Upon receipt of the last pulse expected from the pulse train, the centroid direction of arrival is calculated and the signal information is zeroed in the New Signal Pointer File as well as in the first five input buffer locations. The time of arrival of the last pulse expected is computed utilizing the scan time for one receiver antenna bandwidth.

Non-stable signals are flagged as such and remain in the input buffer. Upon completion of incrementation through the input buffer, all stable pulses, except those from trains that remain due to PTS capacity having been exceeded, have been zeroed from the input buffer. Parameters associated with these trains are stored in sequence according to pulse repetition interval.

The pulse information remaining in the input buffer, associated with the unstable pulse trains and with pulse trains exceeding the PTS capacity, is then compressed within the input buffer, thus allowing for simulation of the mini computer signal processing algorithm. In the initial design concept of this simulation program, these remaining data, representing the Pulse Train Separator residual channel, would be processed by an ATAC assembly language program incorporated as a local procedure within the PTS program using the code selection head as described in the PL/ATAC manual.

A complete PL/ATAC program listing of the Pulse Train Separator is contained in Appendix C. An assembly language



program generated by the PL/ATAC compiler is contained in Appendix D. A program flowchart is illustrated in Figures 2 and 2a.



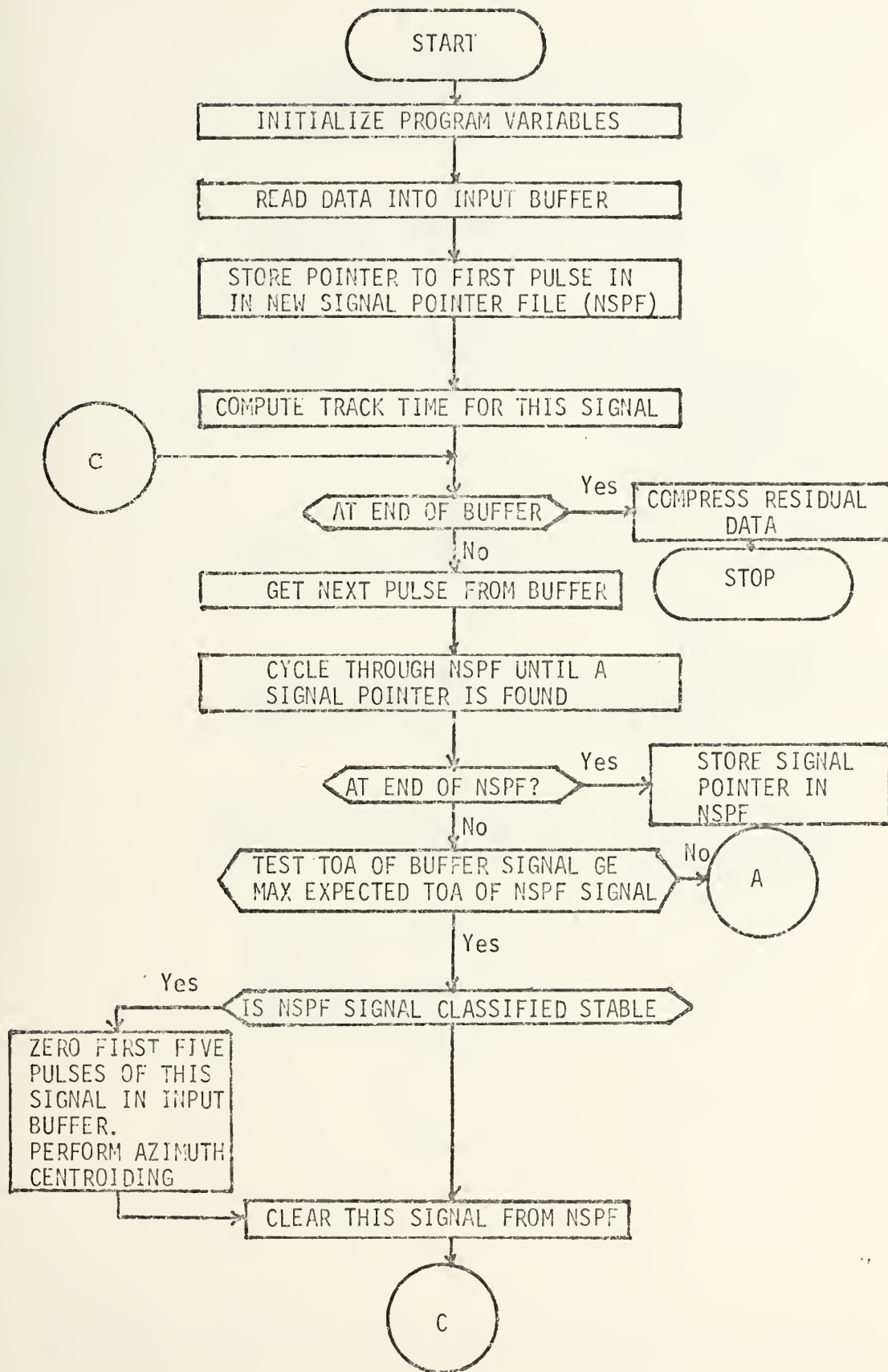


FIGURE 2. The Pulse Train Separator Program Flowchart



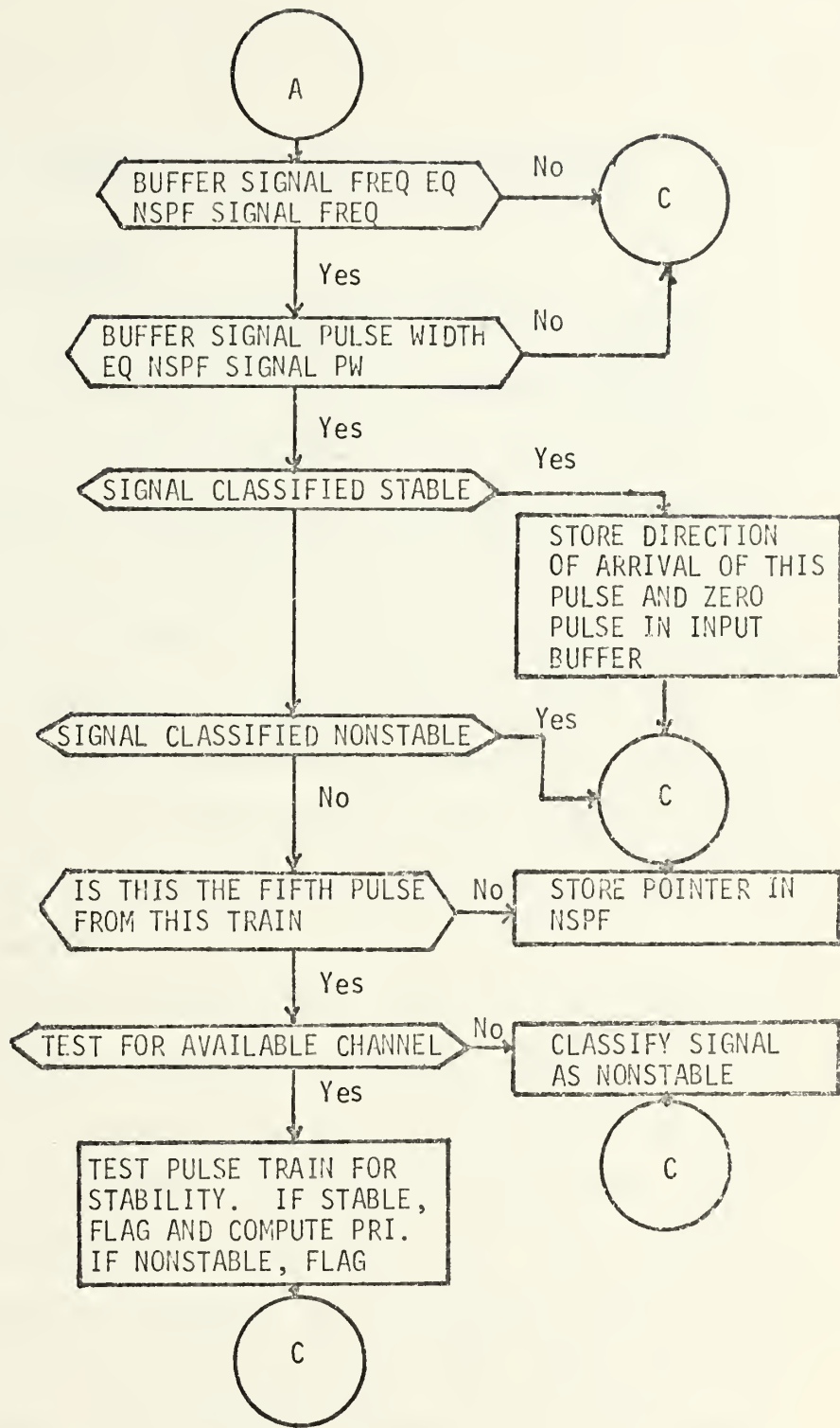


FIGURE 2a. The Pulse Train Separator Program Flowchart



## V. DISCUSSION OF SIMULATION RESULTS

### A. SUMMARY OF RESULTS

The Pulse Train Separator as described in Chapter IV was simulated utilizing the ATAC Programming Support System and the PL/ATAC Programming Language. The particular hardware system simulated was arbitrarily given the capability of deinterleaving and preprocessing up to nine radar pulse trains.

Input data, provided by the Data Simulation Program, consisted of a mixture of stable and non-stable pulse trains from eighteen emitters located on six platforms geographically positioned so as to be intercepted in one receiver antenna beamwidth. The receiver antenna beamwidth was arbitrarily set at five degrees.

Using a receiver antenna scan rate of two seconds per revolution, the Data Simulation Program generated a total of 365 pulses as IFM output during one antenna main beam sweep of the eighteen emitters. These pulses represent approximately 28 milliseconds of real time data. The Pulse Train Separator Simulation Program, in processing nine of the pulse trains, removed 277 pulses from the input data and routed 88 pulses through the residual channel. These 88 pulses represent the unstable pulse trains and the lower pulse repetition frequency (PRF) signals. Accurate pulse repetition interval calculations as well as accurate azimuth centroiding was performed on all preprocessed pulse trains.



## B. CONCLUSIONS

The Data Simulation Program provides realistic data to the APSS programs. The program produces reasonably complex data in the form of jittered pulse trains and staggered pulse trains. Pulse trains transmitted at sufficiently low power levels are produced with missing pulses due to  $\sin(x)/x$  effects.

The Pulse Train Separator Simulation Program demonstrated the use of the PL/ATAC language and functions as designed to preprocess ESM signals. This program is somewhat inefficient in its present form for two reasons. First, the limitation of five registers allocated to the programmer by the PL/ATAC compiler leads to a significant increase in program execution time as compared to a program written in ATAC assembly language. As the programmer gains experience with the PL/ATAC language, this limitation of the compiler may be overcome somewhat by programmer 'tricks' to gain use of registers.

The lack of a double precision capability in the current version of the PL/ATAC compiler installed on the IBM 360/67 at the Naval Postgraduate School created a severe handicap in writing the Pulse Train Separator Simulation Program and also contributes to program inefficiency. Each double precision calculation performed requires a program segment of assembly language and the housekeeping involved results in increased program execution time.

Even with these limitations it is felt that the simulation programs presented in this work provide the framework for ESM



software development and applications programming at the Naval Postgraduate School.

### C. RECOMMENDATIONS

Based upon the experience and results obtained from this study, recommendations for follow-on study in the area of ESM software development are as follows:

1. Continue study of the Pulse Train Separator Program and optimize the basic algorithm.
2. Implement the resulting improved algorithm in ATAC assembly language.
3. Use the Data Simulation Program and optimized hardware simulation with APSS to develop and optimize ESM signal processing algorithms.
4. Continue use of the PL/ATAC compiler to develop and test hardware simulation and other programs prior to implementation in ATAC assembly language.



## APPENDIX A

### INSTRUCTIONS FOR INSTALLATION AND USE OF THE ATAC PROGRAMMING SUPPORT SYSTEM (APSS) ON THE IBM 360/67 OPERATING SYSTEM AT THE NAVAL POSTGRADUATE SCHOOL

#### A. INSTALLATION

The ATAC Programming Support System is currently loaded on the 2321 Data Cell. Data set names and their locations are as follows:

1. S1615.RDS.LOADLIB - CEL 007
2. S1616.RDS.APSS - CEL 005
3. S1615.RDS.PLATAC - CEL 001

In the event of failure of the Data Cell these data sets may be reloaded from the APSS source tape stored at the W.R. Church Computer Center under the name 'ATI.006'.

File #1 of this tape contains a description of the tape. File #2 contains examples of the JCL required to create and run APSS. The instructions contained in this file are superseded by the instructions contained here. The following JCL is required to list these files:

```
//(STANDARD JOB CARD)
//EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT2 DD SYSOUT=A, DCP=(LRECL=80,RECFM=F,BLKSIZE=80)
        LABEL=(1,SL,,IN),DCB=(BLKSIZE=800,LRECL=80,
        RECFM=FB),VOL=SER=ATI006,DSN=FILE1
// EXEC PGM=IEBGENER
// SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT2 DD SYSOUT=A,DCB=(LRECL=89,RECFM=F,BLKSIZE=80)
//SYSUT1 DD UNIT=(2400,,DEFER),DISP=(OLD,PASS),
        LABEL=(1,SL,,IN),DCB=(BLKSIZE=800,LRECL=80),
        RECFM=FB,VOL=SER=ATI006,DSN=FILE2
```



File #3 of the source tape contains the object module library of APSS and is required to build the APSS Instruction Level Simulator. File #4 contains the PL/ATAC Compiler object module. The following steps are required to build APSS from files 3 and 4.

1. Read files 3 and 4 from tape to disc storage. Special arrangements should be made to insure that the disc pack used has sufficient unallocated space for this step and the following step of this procedure. SPOOL3 has been made available in the past and is the disc used in the JCL of this example. The load modules from file 3 are moved into a disc file named 'S1614.RDS.LOADLIB'. File 4 is moved into a disc file named 'S1615.RDS.PLATAC'. The utility control statements may be changed to rename the files as they are being loaded onto the disc. The JCL statements required are as follows:

```
//(STANDARD JOB CARD)
//STEP1 EXEC PGM=IEHMOVE, REGION=LOOK
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=SYSDA,SPACE=(TRK,(40),,CONTIG)
//DA1 DD UNIT=2314,DSN=S1615.RDS.LOADLIB,
//SPACE=(TRK,(50,10,10),,CONTIG),
//DISP=(NEW,KEEP),VOL=SER=SPOOL3
//T1TAPE DD UNIT=(2400,,DEFER),DISP=(NEW,PASS),
//      LABEL=(3,SL,,IN),
//DCB=(DEN=2,BLKSIZE=800,LRECL=80,RECFM=FB),
//      VOL=SER=ATI006
//SYSIN DD *
COPY PDS=ATI.APSS.LOADLIB,TO=2314=SPOOL3,TODD=DA1,
//      FROMDD=T1TAPE,*
FROM=2400=(ATI006,3),RENAME=S1615.RDS.LOADLIB
COPY DSN=ATI.APSS.PLATAC,TO=2314=SPOOL3,
//      FROMDD=T1TAPE,TODD=DA2,*
FROM=2400=(ATI006,4),RENAME=S1615.RDS.PLATAC
//DA2 DD UNIT=2314,DSN=S1615.RDS.PLATAC,SPACE=(TRK,
//      (20,10),RLSE),
//DISP=(NEW,KEEP),VOL=SER=SPOOL3,LABEL=RETPD=360
/*
```



2. The Library (named 'S1615.RDS.LOADLIB') now on the disc must be built into an absolute load module of the entire APSS system. The module thus built (named 'S1615.RDS.APSS') will be stored permanently on cell 005 of the data cell. This file may also be renamed as desired. The following JCL statements are required.

```
//(STANDARD JOB CARD)
//BUILD EXEC PGM=IEWL,REGION=150K,
//PARM='OVLY,XREF,LET,LIST,SIZE=(256K,20480)'
//SYSPRINT DD SYSOUT=A
//LIBRARY DD DSN=S1615.RDS.LOADLIB,UNIT=2314,
           VOL=SER=SPOOL3,DISP=SHR
//SYSLIB DD DSNAME=3YS1.FORTLIB,DISP=SHR
//SYSLMOD DD DSNAME=S1615.RDS.APSS,UNIT=2321,
           VOL=SER=CEL005,
//DISP=(NEW,KEEP),LABEL=RETPD=360,
//SPACE=(TRK,(300,20,2),RLSE)
//SYSUT1 DD UNIT=SYSDA,SPACE=(TRK,(19,19),,CONTIG),
           SEP=SYSLMOD
//SYSLIM DD*
INCLUDE LIBRARYEPMUDLN
CHANGE MSIMEIHESAPDN
INCLUDE LIBRARY (APSSMON)
INCLUDE LIBRARY (MSIM4,ASEM5,SIM16A,SIMTR1,SIMI01,GUL)
INCLUDE LIBRARYEXPLMONN
OVERLAY A1
INSERT MSIMUL,*MSIMULA,IHENTRY,IHESAP
INSERT MINT, IN, OUT
INSERT IHEDBN, IHEXID
INSERT IHEBSM, IHECSM
INSERT IHEBSK, IHEIOX, IHEIOP, IHEDID, IHEDOB
INSERT IHEBIB, IHEDOA, IHEIOB
INSERT IHEIOA, IHEOCL
INSERT IHEBSD, IHEBSF
INSERT IHEJXS
INSERT IHEOSD, IHEOST, IHEBSI
INSERT IHEVPE, IHEDMA, IHEVFB
INSERT IHEDNC, IHEVFD, IHEVFA, IHEVPD, IHEVPB, IHEVSC
INSERT IHEVSD, IHEVFE, IHEDCN, IHEUPB
INSERT IHEVFC, IHEVPE, IHEVPG, IHEVQB, IHEVQC
INSERT IHEABN, IHEIOD, IHEIOF, IHEPRT, IHEVQA, IHESPRT
INSERT IHEBEG, IHEERR, IHESI2
INSERT MISET
OVERLAY A1
INSERT ASSEM, REWIND, REW72, DSKOUT, CARDIN, DISKIN,
           ERPRT, PRIADD
```



```

INSERT WRDATA, PRICOM, PRINOR, WRITEX, REFTIT, PREF,
      ERTIT
OVERLAY A1
INSERT PARMRD, PRESIM
OVERLAY A1
INSERT SMLTR, NRMTRM, STRTSM, TPAGE
INSERT RDCRD, ABNRMT, ARTHFP, TRACE, HGRAM, HGRAMI,
      HGRAMS
INSERT IOINIT, ACTIVE, SIMTIM
INSERT DEVDTA, ACT, TIME, INT, RAND, DEADT, DEBUG
OVERLAY A2
INSERT LEVEL, DMAIOI, DMAIOA, DMAIOD, RIOIO
INSERT REMACT, DMA, DMATM, RIO, RIOTM, RIOINT, INTOLY,
      DMAINT
INSERT DTRAN, PUTACT, RANDOM
OVERLAY A1
INSERT HGPRNT
OVERLAY A1
INSERT LAODER
OVERLAY A1
INSERT LINK, ENTEXT, SLH
OVERLAY A1
INSERT PLATAC, IOPACK
OVERLAY $OBJECTEREGIONN
INSERT OBJECT, INIT, LIB, RCALPH, RCHEX, RCINT
INSERT MDATE
OVERLAY $DUMPEREGIONN
INSERT SMDUMP,PAGE
ENTRY MAIN
NAME APSS
/*

```

3. The PL/ATAC compiler object module on the disc must be moved to the data cell.

```

//(STANDARD JOB CARD)
//EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DISP=SHR,UNIT=2314,VOL=SER=SPOOL3,
//DSN=S1615.RDS.PLATAC
//SYSUT2 DD DISP=(NEW,KEEP),UNIT=2321,VOL=SER=CEL001,
//DCB=(RECFM=FT,BLKSIZE=7188),LABEL=RETPD=360,
//SPACE=(7188,(20,0)),DSN=S1615.RDS.PLATAC
//SYSIN DD *
GENERATE
/*

```



4. The Library ('S1615.RDS.LOADLIB') must now be moved from the disc to the data cell for permanent storage. This library contains modules necessary for successful execution of the PL/ATAC compiler. The required JCL statements to accomplish this step are as follows:

```
//(STANDARD JOB CARD)
//EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DISP=SHR,UNIT=2314,VOL=SER=SPOOL3,
//DSN=S1615.RDS.LOADLIB
//SYSUT2 DD DISP=(NEW,KEEP),UNIT=2321,VOL=SER=CEL007,
//DSW=S1615.RDS.LOADLIB,SPACE=(7294,(61,0,14)),
//DCB=(RECFM=UT,BLKSIZE=7294),LABEL=RETPD=360
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(20,5))
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(20,5))
//SYSIN DD *
COPY OUTDD=SYSUT2,IWDD=SYSUT1
/*
```

#### B. USER INSTRUCTION

The JCL required to compile and execute PL/ATAC programs is as follows:

```
//(STANDARD JOB CARD)
//PLATAC EXEC PGM=XPLMON,PARM='ALTER',REGION=200K
//STEPLIB DD DSN=S1615.RDS.LOADLIB,UNIT=2321,DISP=SHR,
VOL=SER=CEL007
//PROGRAM DD DSN=S1615.RDS.PLATAC,UNIT=2321,VOL=SER=CEL001,
DISP=SHR
//SYSPRINT DD SYSOUT=A
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//FT27F001 DD DSN=&&PLCODE,SPACE=(TRK,(20,10),RLSE),
//DCB=(BLKSIZE=3520,LRECL=80,RECFM=FB),DISP=(NEW,PASS,DELETE),
//UNIT=SYSDA
//SYSIN DD *
```

⋮

PL/ATAC Program

⋮



```

//APSS EXEC PGM=APSS REGION=200K
//STEPLIB DD DSN=S1615.RDS.APSS,UNIT=2321,DISP=(OLD,KEEP) ,
//      VOL=SER=CEL005,DCB=BLKSIZE=400
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3325)
//FT08F001 DD UNIT=SYSDA,SPACE=(CYL,1),DCB=BUFNO=1
//FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(7,2)),
//      DCB=(RECFM=VBS,BLKSIZE=7180,LRECL=92,BUFNO=1)
//FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(7,2)),
//      DCB=(RECFM=VBS,BLKSIZE=4204,LRECL=42,BUFNO=1)
//FT20F001 DD UNIT=SYSDA,SPACE=(CYL,(7,2)),
//      DCB=(BUFNO=1,RECFM=VBS,BLKSIZE=2004,LRECL=500)
//FT27F001 DD DSN=&&PLCODE,UNIT=SYSDA,DISP=(OLD,DELETE) ,
//      DCB=BUFNO=1
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3325)
//SYSIN DD *
$END
/*

```



## APPENDIX B

### DATA SIMULATION PROGRAM FORTRAN LISTING AND OUTPUT

A description of program variables is included in the Data Simulation Program listing. The input data deck setup is also included in the listing.



















|                   |         |        |          |                              |  |      |
|-------------------|---------|--------|----------|------------------------------|--|------|
| INPUT CARD FORMAT |         |        |          | EMITTER PLATFORM INFORMATION |  |      |
| CARD              | COLUMNS | FORMAT | VARIABLE | UNITS                        |  |      |
| 1                 | 1-10    | F10.0  | XI       | NAUTICAL MILES               |  | 1830 |
|                   | 11-20   | F10.0  | YI       | NAUTICAL MILES               |  | 1840 |
|                   | 21-30   | F10.0  | ZI       | FEET                         |  | 1850 |
|                   | 31-40   | F10.0  | CSEI     | DEGREES CLOCKWISE FROM NORTH |  | 1870 |
|                   | 41-50   | F10.0  | SPDI     | DEGREES                      |  | 1890 |
|                   | 51      | I1     | NMTR(I)  | KNOTS                        |  | 1900 |
|                   |         |        |          | INTEGER                      |  | 1910 |
| INPUT CARD FORMAT |         |        |          | EMITTER INFORMATION          |  | 1920 |
| CARD              | COLUMNS | FORMAT | VARIABLE | UNITS                        |  | 1930 |
| 1                 | 1-10    | F10.0  | FREQ     | MHZ                          |  | 1940 |
|                   | 11-20   | F10.0  | PRI      | MILLISECONDS                 |  | 1950 |
|                   | 21-30   | F10.0  | PW       | MICROSECONDS                 |  | 1960 |
|                   | 31-40   | F10.0  | PYTRANS  | KILOWATTS                    |  | 1970 |
|                   | 41-50   | F10.0  | BEAM     | DEGREES                      |  | 1980 |
|                   | 51-60   | F10.0  | GAIN     | DB                           |  | 1990 |
|                   | 61-70   | F10.0  | SCAN     | SECONDS PER REVOLUTION       |  | 2000 |
|                   | 1-10    | F10.0  | PJTI     | REAL NUMBER PERCENT          |  | 2010 |
| 2                 | 11-20   | F10.6  | STGRI    | REAL NUMBER PERCENT          |  | 2020 |
| END OF DATA       |         |        |          | NEGATIVE NUMBER IN CC 21-30  |  | 2030 |
|                   |         |        |          |                              |  | 2040 |
|                   |         |        |          |                              |  | 2050 |
|                   |         |        |          |                              |  | 2060 |
|                   |         |        |          |                              |  | 2070 |
|                   |         |        |          |                              |  | 2080 |
|                   |         |        |          |                              |  | 2090 |



```

COMMON X(10),Y(10),Z(10),VX(10),VY(10),BNG(10),F(10,5),TINT,VXO,VY
10,THRESH,PI,PI2,SCNMIN,PRIN,SIGFRQ(10,5),PRI(10,5),PW(10,5),POWER(
210,5),BW(10,5),SEND(10,5),TNEXTI(10,5),SIGNAL(10,5),ANT(10,5),ANG(1
30,5),XQ,YQ,ZO,CSEQ,SPDQ,VL,TWOPI,RDEGR,TO,TF,FMAX,FMIN,BIN,PI,X
41F(10,5),PJT(10,5),STGR(10,5),IPT(10,5),ISCAN(10,5),JNEX
51(10),NUNI,NGAUS,NMTR(10),TFINI(10,5),RSPR,RRATE,RBW,RGAIN,NOUT(40
6)

```

CCCCC

#### INITIALIZE CONSTANTS

```

PI2 = ARSIN(1.0)
PI = 2.*PI2
TWOPI = 40.*PI2
RDEGR = 180./PI
DEGR = PI/180.
CON1 = PI*58.9*DEGR
CON2 = 10.*ALOG10(4.*PI)
TSKTS = 2025.4
YDS = 0.562611
FSCALE = 1.E6
TSCALE = 1.E3
WSCALE = 1.E-3
IJ = 0
ICT = 0

```

CCCCC

#### SET IFM INFORMATION

```

READ (5,17) NUNI,NGAUS,VL,TO,TF,TINT,THRESH
READ (5,18) XQ,YQ,ZO,CSEQ,SPDQ,FMAX,FMIN
READ (5,21) NBIN,RBW,RSPR,RGAIN
VL = VL*FSCALE
TINT = TINT*WSCALE
FMAX = FMAX*FSCALE
FMIN = FMIN*FSCALE
BIN = (FMAX-FMIN)/FLOAT(NBIN)

```

CCCCC

#### SET IFM INITIAL ANTENNA POSITION

```

CALL RANDU (NUNI,IV,RAND)
NUNI = IV
RANT = TWOPI*RAND

```



```

CCCCC
RRATE = TWOPI/RSRPR
OUTPUT IFM INFORMATION
WRITE (6,23) XJ,YO,ZO,CSEO,SPDO,FMIN,FMAX,BIN,NBIN
WRITE (6,24) THRESH
WRITE (6,30) RGAIN
WRITE (6,31) RSPR
CSEO = CSEO*DEGR
SPDO = SPDO*TSKTS
XO = XO*YDS
YO = YO*YDS
ZO = ZO/3.
VXO = SPDO*SIN(CSEO)
VYO = SPDO*COS(CSEO)
I = 1

GET EMITTER PLATFORM INFORMATION

1 READ (5,19) XI,YI,ZI,CSEI,SPDI,NMTR(I)

IF ALL EMITTERS HAVE BEEN ENTERED GO SORT TOA'S

IF (ZI,LT,0.) GO TO 10
WRITE (6,25) I,XI,YI,ZI,CSEI,SPDI

COMPUTE PLATFORM RELATIVE COURSE AND SPEED

CSEI = CSEI*DEGR
SPDI = SPDI*TSKTS
VXI = SPDI*SIN(CSEI)-VXO
VYI = SPDI*COS(CSEI)-VYO
XI = XI*YDS-XO
YI = YI*YDS-YO
Z(I) = SQRT(VXI**2+VYI**2)
VTI = ATAN2(VXI,VYI)
RCSE = ATAN2(VXI,VYI)

COMPUTE RANGE AND BEARING TO PLATFORM
CC

```



```

CC
CCCCC
    R = SQR T(XI**2+YI**2)
    BNGI = ATAN2(XI,YI)

    COMPUTE PULSE TIME OF FLIGHT FOR THIS PLATFORM

    TOF = R/VL
    X(I) = XI
    Y(I) = YI
    VX(I) = VXI
    VY(I) = VYI
    BNG(I) = BNGI

    COMPUTE CPA AND DISTANCE TO CPA

    THETA I = ABS(PI-BNGI+RCSE)
    IF (THETA I.GT.TWOPI) THETA I=THETA I-TWOPI
    IF (THETA I.LE.PI2) GO TO 2
    DI = R*COS(THETA I)
    CPA = R*SIN(THETA I)
    GO TO 3
2 CPA = R
3 JJ = NMTR(I)

    COMPUTE TIME WINDOW IN WHICH THIS PLATFORM WILL BE IN THE
    RECEIVER ANTENNA BEAMWIDTH

    IF (BNGI.LE.0.) BNGI=BNGI+TWOPI
    IF (ABS(BNGI-RANT).LE.((RBW/2.)*DEGR)) GO TO 4
    ANGR I = BNGI-RBW/2.*DEGR-RANT+TWOPI
    IF (ANGRI.LT.0.) ANGR I=ANGRI+TWOPI
    ANGRF = BNGI+RBW/2.*DEGR-RANT
    IF (ANGRF.LT.0.) ANGRF=ANGRF+TWOPI
    TINI = ANGRI/RRATE+TO
    GO TO 5
4 TINI = TO
5 TFIN = ANGRF/RRATE+TO

    GET EMITTER INFORMATION

```



```

CC          CC          CC          CC          CC          CC          CC          CC          CC          CC
DO 9 J=1,JJ      FREQ,PRIN,PULSE,PTRANS,BEAM,GAIN,ANT(I,J)
READ (5,18)
READ (5,20) PJTI,SIGRI
3540
3550
3560
3570
3580
3590
3600
3610
3620
3630
3640
3650
3660
3670
3680
3690
3700
3710
3720
3730
3740
3750
3760
3770
3780
3790
3800
3810
3820
3830
3840
3850
3860
3870
3880
3890
3900
3910
3920
3930
3940
3950
3960
3970
3980
3990
4000
4010

SET PARAMETERS

PJT(I,J) = PJTI
SIGRI(I,J) = SIGRI
PTRANS = PTRANS*TSSCALE
F(I,J) = FREQ*FSSCALE
PRIN = PRIN*PSCALE
PRI(I,J) = PRIN
PULSE = PULSE*WSSCALE
PW(I,J) = PULSE
TFINI(I,J) = TFIN
BW(I,J) = CON1/BEAM

COMPUTE EFFECTIVE RADIATED POWER

POWIJ = 10.*ALOG10(PTRANS)+GAIN-20.*ALOG10(FREQ)+58.3335
POWER(I,J) = POWIJ

OUTPUT PARAMETERS OF THIS EMITTER

WRITE (6,26) J,F(I,J),PRI(I,J),PW(I,J),PTRANS,GAIN,BEAM,POWER(I,J)
WRITE (6,27) ANT(I,J)

COMPUTE MAX RANGE OF DETECTION

RMAX = 10.*((POWIJ+RGAIN-THRESH)/20.)

CALCULATE THE TOA OF THE FIRST PULSE

IF (RMAX.GE.CPA) GO TO 6
TNEXT(I,J) = TFF+1.
WRITE (6,28)

```



```

4020 GO TO 9
4030 CALL RANDU (NUNI,IV,RAND)
4040 NUNI = IV
4050 ANG(I,J) = RAND*TWOPI
4060 CALL RANDU (NUNI,IV,RAND)
4070 NUNI = IV
4080
4090 IF EMITTER RANGE GREATER THAN MAX RANGE OF DETECTION GO COMPUTE
4100 DELAY TIME
4110 OTHERWISE GO COMPUTE A TRANSMIT TIME WITHIN THE EMITTER
4120 PRI OF THE TIME EMITTER WILL BE IN THE RECEIVER ANTENNA BEAM
4130
4140
4150
4160
4170
4180
4190
4200
4210
4220
4230
4240
4250
4260
4270
4280
4290
4300
4310
4320
4330
4340
4350
4360
4370
4380
4390
4400
4410
4420
4430
4440
4450
4460
4470
4480
4490

```

```

6 GO TO 9
  CALL RANDU (NUNI,IV,RAND)
  NUNI = IV
  ANG(I,J) = RAND*TWOPI
  CALL RANDU (NUNI,IV,RAND)
  NUNI = IV

  IF EMITTER RANGE GREATER THAN MAX RANGE OF DETECTION GO COMPUTE
  DELAY TIME
  OTHERWISE GO COMPUTE A TRANSMIT TIME WITHIN THE EMITTER
  PRI OF THE TIME EMITTER WILL BE IN THE RECEIVER ANTENNA BEAM

  IF (RMAX.LT.R) GO TO 7
  DEL = 0.
  R1 = R
  GO TO 8
7 D2 = CPA*SIN(ARCOS(CPA/RMAX))
  DEL = (D1-D2)/VTI+(R-RMAX)/VL
  R1 = RMAX
  TT = TINI+PRIN*RAND-TOF-PULSE+DEL
  IF ((TT+TOF+DEL).LT.TINI) TT=TT+PRIN

  COMPUTE RANGE ATTENUATION

  POW = POWIJ-20.*ALOG10(R1)

  COMPUTE MINIMUM EMITTER ANTENNA GAIN REQUIRED FOR RECEPTION

  SCNMN = THRESH-POW-RGAIN

  COMPUTE RECEIVED SIGNAL POWER LEVEL
  IF ANTENNA GAIN (COMPUTED IN SCAN) WAS BELOW THE REQUIRED VALUE
  FUNCTION SCAN WILL HAVE INCREMENTED TIME AT THE EMITTER PRI
  UNTIL EMITTER ANTENNA GAIN AT SUFFICIENT LEVEL OR TIME EXCEEDS
  SIMULATION STOP TIME

  SIGNAL(I,J) = POW+SCAN(I,J)

  COMPUTE TOA

```

CCCCCCCC

CCCCC CCCCC CCCCCCCCCC C



```

CC      TNEXT(I,J) = TINT*AINI((TT+TOF)/TINT)
CC      SEND(I,J) = TT
CC      SIGFRO(I,J) = F(I,J)
CC      9 CONTINUE
CC      GO GET NEXT PLATFORM INFO
CC      I = I+1
CC      GO TO I
CC      AT THIS POINT ALL OF THE EMITTER PLATFORMS AND ALL EMITTERS
CC      HAVE BEEN READ IN AND THE TOA OF THE FIRST PULSE OF EACH
CC      EMITTER HAS BEEN COMPUTED
CC      ROUTINE BELOW IS TO FIND THE TOA WITH THE LOWEST VALUE
CC
CC      10 NTGT = I-1
CC      WRITE (6,29)
CC      11 TEMP = IF
CC      L = 0
CC      ITEMP = 0
CC
CC      DO 14 I=1,NTGT
CC      JJ = NMTR(I)
CC
CC      DO 13 J=1,JJ
CC      TTRY = TNEXT(I,J)
CC      IF (TEMP.LT.TTRY) GO TO 13
CC      IF (TEMP.EQ.TTRY) GO TO 12
CC      ITEMP = TTRY
CC      JTEMP = J
CC      GO TO 13
CC      12 L = L+1
CC      INEXT(L) = ITEMP
CC      JNEXT(L) = JTEMP
CC      ITEMP = I
CC      JTEMP = J
CC      13 CONTINUE
CC

```



```

C      14 CONTINUE
C
C      IF (ITEMP.EQ.0) STOP
C
C      CONVERT PULSE DATA TO FORMAT REQUIRED BY THE ATAC COMPUTER
C      AND OUTPUT DATA ON PRINTER AND CARDS
C
15  RCVSIG = SIGNAL(ITEMP,JTEMP)
    RCVR = SIGFRQ(ITEMP,JTEMP)
    N = (RCVR-FMIN)/BIN+1.
    ICT = ICT+1
    RPW = 1.E7*PW(ITEMP,JTEMP)
    NPW = RPW
    IF ((RPW-NPW).GT.5) NPW=NPW+1
    TNEXT1 = TNEXT(ITEMP,JTEMP)
    RCVBNG = AMOD((RANT+TNEXT1*TWOPI/RSPR),TWOPI)*RDEG
    IRXB = RCVBNG
    NTIME1 = TNEXT1*1.E7/256.
    NTIME1 = MOD(NTIME,65536)
    NTIME2 = NTIME/65536
    WRITE (6,22) IRXB,ICT,N,NTIME2,NTIME1,NPW,TNEXT1
    IF (TNEXT1.GT.1F) STOP
    NOUT(IJJ+1) = IRXB
    NOUT(IJJ+2) = NTIME1
    NOUT(IJJ+3) = NTIME2
    NOUT(IJJ+4) = NTIME2
    NOUT(IJJ+5) = NPW
    IJJ = IJJ+5
    IF (IJJ.LT.40) GO TO 16
    IJJ = 0
    WRITE(7,32)NOUT
16  IF (ICT.GE.800) STOP
C
C      GO CALCULATE THE TOA OF THE NEXT PULSE FROM THIS EMITTER
C
C      CALL UPDATE (ITEMP,JTEMP)
C      IF (L.EQ.0) GO TO 11
C      ITEMP = INEXT(L)
C      JTEMP = JNEXT(L)
C      L = L-1
C      GO TO 15
C

```







SUBROUTINE UPDATE (I,J)

UPDATE IS CALLED FOR CALCULATION OF ALL TOAS AFTER THE FIRST  
PULSE FOR EACH EMITTER. HERE THE EMITTER IS FIRST TESTED FOR  
JITTER CHARACTERISTICS. IF JITTER PRESENT COMPUTE RANDOM  
PULSE INTERVAL IN JITTER BOUNDS  
IF NO JITTER CHARACTERISTICS TEST FOR STAGGER. IF STAGGER  
PRESENT COMPUTE PULSE INTERVAL AT PROPER STAGGER RATIO  
IF NO STAGGER NORMAL CALCULATIONS ARE PERFORMED  
PLATFORM RANGE AND BEARING ARE UPDATED  
POWER AND TOA CALCULATIONS ARE PERFORMED

COMMON X(10), Y(10), Z(10), VX(10), VY(10), BNG(10), F(10,5), TINT, VXO, VY  
10, THRESH, PI, PI2, SCNMN, PRIN, SIGFRQ(10,5), PRI(10,5), PW(10,5), POWER(1  
210,5), BW(10,5), SEND(10,5), TNEXT(10,5), SIGNAL(10,5), ANT(10,5), ANG(1  
310,5), XO, YO, ZO, CSEQ, SPDO, VL, TWOP, I, RDEG, DEGR, TO, IF, FMAX, FMIN, BI, TI,  
41F(10,5), PJT(10,5), STGR(10,5), IPT(10,5), ISCAN(10,5), INEXT(10), JNEX  
51PRIN = PRI(I,J), NGAU, NMTR(10), TFINI(10,5), RSPR, RRATE, RBW, RGA

IF (PJT(I,J).EQ.0) GO TO 1  
PRICOM = PRI(I,J) - PJT(I,J) \* PRI(I,J) / 2.

CALL RANDU (NUNI, IV, RAND)

NUNI = IV  
PRIN = PRICOM + RAND \* PJT(I,J) \* PRI(I,J)

GO TO 4  
1 IF (STGR(I,J).EQ.0.) GO TO 3

IF (STGR.EQ.1) GO TO 2

PRIN = PRI(I,J) - (1. - STGR(I,J)) \* PRI(I,J)

GO TO 4

2 STGR = 0  
PRIN = PRI(I,J) - STGR(I,J) \* PRI(I,J)

GO TO 4

3 PRIN = PRI(I,J)  
4 TFINI = TFINI(I,J) + PRIN

PRIN = SEND(I,J) \* PRIN  
IF (TFINI) GO TO 5

DTFIN = (TFINI - RBW \* DEGR / RRATE  
IFIN = TFINI + RSPR

5 TFINI = TFINI + (AINT(DT / PRIN) + 1.) \* PRIN  
XFI = X(I)

YFI = Y(I)

VXI = VX(I)

VYI = VY(I)

XX = XI + VXI \* TT



```

YY = YI+VYI*TT
R = SQRT(XX**2+YY**2)
POW = POWER(I,J)-20.*ALOG10(R)
SCNMIN = THRESH-POW-RGAIN
IF (SCNMIN.GE.0.) GO TO 6
SIGNAL(I,J) = POW+SCAN(I,J)
TOF = R/VL
BNG(I) = ATAN2(XX,YY)
TNEXT(I,J) = TINT+AINI*((TT+TOF)/TINT)
SEND(I,J) = TT
RETURN
6 TNEXT(I,J) = IF+1.
WRITE (6,7) I,J
RETURN
C
7 FORMAT (' EMITTER',2I3,' OUT OF RANGE.')
END

```

```

6170
6180
6190
6200
6210
6220
6230
6240
6250
6260
6270
6280
6290
6300
6310
6320
6330
6340

```



FUNCTION SCAN (I,J)

SCAN COMPUTES THE RELATIVE ANGLE OF THE EMITTER ANTENNA AT  
TRANSMIT TIME. SIN X /X ATTENUATION IS COMPUTED  
IF ANTENNA GAIN IS AT OR ABOVE REQUIRED LEVEL THE PROGRAM  
RETURNS TO CALLING POINT WHERE TOA WILL BE COMPUTED.  
IF ANTENNA GAIN IS TO LOW TO BREAK RECEIVER THRESHOLD, TIME  
IS INCREMENTED AT THE PRI RATE UNTIL SUCH TIME AS A  
SIGNAL BREAKS THRESHOLD

COMMON X(10),Y(10),Z(10),VX(10),VY(10),BNG(10),F(10,5),TINT,VXD,VY  
10,THRESH,P1,P12,SCNMIN,PRIN,SIGFRQ(10,5),PRI(10,5),PWT(10,5),POWER(1  
2:0,5),BW(10,5),SEND(10,5),TNEXT(10,5),SIGNAL(10,5),ANT(10,5),ANG(1  
3:0,5),XQ,YO,ZQ,CSEQ,SPDO,VL,TWOPI,RDEG,DEGR,TO,TF,FMAX,FMIN,BIN,FI,  
4:1F(10,5),PJ(10,5),STGR(10,5),IPI(10,5),ISCAN(10,5),INEXT(10),JNEX  
5:1(10),NUN,NGAUS,NMTR(10),TFINI(10,5),RSPR,RRATE,RBW

1 ANGI = ANG(I,J)  
IFIN = TFIN(I,J)

BWI = BW(I,J)

BNGI = BNG(I,J)

RATE = ANTI(I,J)

RATI = TWOPI/RATE

ANGLE = AMOD((ANGI+TT\*RATI-PI-BNGI),TWOPI)

IF (ANGLE.GT.PI) ANGLE=ANGLE-TWOPI

IF (ANGLE.LT.(-PI)) ANGLE=ANGLE+TWOPI

IF (ABS(ANGLE).LE.1.E-2) GO TO 5

U = BWI\*ANGLE

SCAN = 10.\*ALOG10((SIN(U)/U)\*\*2)

IF (SCAN.GE.SCNMIN) GO TO 4

TT = TT+PRIN

IF (TT.GE.TFIN) GO TO 3

GO TO 2

3 DTFIN = TFIN+RSPR-RBW\*DEGR/RRATE

DTFIN = TFIN+RSPR

TFINI(I,J) = TFIN

TT = TT+(AINT(DT/PRIN)+1.)\*PRIN

IF (TT.GE.TF) GO TO 4

GO TO 2

4 RETURN 0.0

5 RETURN

END

CCCCCCCCCCCC



OWN UNIT INFORMATION.

INITIAL POSITION: 0.0 0.0 0.0  
 THRESHOLD = 70.0 DBM  
 RCVR ANTENNA GAIN = 25.0 DBM  
 RCVR ANTENNA SCAN = 2.00 SPR

TARGET INFORMATION

| PLATFORM           | INITIAL POSITION   | COURSE     | SPEED        | FREQUENCY COVERAGE     | BINSIZE    | NR         | BINS          |
|--------------------|--------------------|------------|--------------|------------------------|------------|------------|---------------|
| 1                  | -98.0 23.0 0.0     | 0.0        | 250.0        | 8.0000E 09- 1.2000E 10 | 3.1250E 07 |            | 128           |
| EMITTER            | FREQ               | PRI        | TRANS. POWER | ANT. GAIN              | BEAM WIDTH | EFF. POWER | ANT. SCAN     |
| 1                  | 9.5950E 09         | 4.0000E-03 | 1.5000E 03   | 2.5000E 01             | 1.0000E 00 | 5.509E 01  | CIR 1.00 SPR  |
| 2                  | 8.8000E 09         | 4.0000E-03 | 2.0000E 03   | 2.0000E 01             | 2.0000E 00 | 3.243E 01  | CIR 8.00 SPR  |
| 3                  | 9.0000E 09         | 2.0000E-03 | 5.0000E 04   | 2.5000E 01             | 5.0000E 00 | 4.9238E 01 | CIR 10.00 SPR |
| TARGET INFORMATION |                    |            |              |                        |            |            |               |
| 2                  | -100.0 24.0 0.0    | 0.0        | 0.           |                        |            |            |               |
| EMITTER            | FREQ               | PRI        | TRANS. POWER | ANT. GAIN              | BEAM WIDTH | EFF. POWER | ANT. SCAN     |
| 1                  | 9.5950E 09         | 4.0000E-03 | 1.5000E 03   | 2.5000E 01             | 1.0000E 00 | 5.5328E 01 | CIR 2.00 SPR  |
| 2                  | 8.5000E 09         | 4.0000E-03 | 1.5000E 03   | 1.5000E 01             | 1.5000E 01 | 4.5328E 01 | CIR 1.00 SPR  |
| 3                  | 9.5000E 09         | 4.0000E-03 | 2.5000E 03   | 1.5000E 01             | 5.0000E 00 | 4.7758E 01 | CIR 9.00 SPR  |
| TARGET INFORMATION |                    |            |              |                        |            |            |               |
| 3                  | -100.0 25.0 0.0    | 0.0        | 15.          |                        |            |            |               |
| EMITTER            | FREQ               | PRI        | TRANS. POWER | ANT. GAIN              | BEAM WIDTH | EFF. POWER | ANT. SCAN     |
| 1                  | 8.5850E 09         | 5.0000E-04 | 1.0000E 05   | 2.0000E 01             | 7.5000E 00 | 5.1360E 01 | CIR 1.00 SPR  |
| 2                  | 1.0147E 10         | 7.5000E-04 | 5.0000E 02   | 3.5000E 01             | 1.5000E 00 | 5.0189E 01 | CIR 0.25 SPR  |
| 3                  | 8.7000E 09         | 1.2500E-03 | 5.0000E 05   | 2.0000E 01             | 2.5000E 01 | 5.6553E 01 | CIR 1.00 SPR  |
| TARGET INFORMATION |                    |            |              |                        |            |            |               |
| 4                  | -100.0 26.0 0.0    | 0.0        | 12.          |                        |            |            |               |
| EMITTER            | FREQ               | PRI        | TRANS. POWER | ANT. GAIN              | BEAM WIDTH | EFF. POWER | ANT. SCAN     |
| 1                  | 9.1880E 09         | 5.0453E-03 | 5.0000E 05   | 1.5000E 01             | 5.0000E 00 | 6.1059E 01 | CIR 0.10 SPR  |
| 2                  | 9.2500E 09         | 1.0000E-03 | 5.0000E 04   | 3.0000E 01             | 5.0000E-01 | 5.6000E 01 | CIR 0.10 SPR  |
| 3                  | 8.6000E 09         | 1.5000E-03 | 5.0000E 04   | 2.5000E 01             | 6.0000E 00 | 5.1633E 01 | CIR 8.00 SPR  |
| 4                  | 0.9000E 09         | 2.0000E-03 | 3.0000E 05   | 1.0000E 01             | 5.0000E 00 | 4.417E 01  | CIR 4.00 SPR  |
| 5                  | 9.6000E 09         | 1.2000E-03 | 3.5000E 05   | 1.5000E 01             | 4.0000E 00 | 4.9119E 01 | CIR 2.00 SPR  |
| TARGET INFORMATION |                    |            |              |                        |            |            |               |
| 5                  | -200.0 50.0 5000.0 | 60.        | 14.          |                        |            |            |               |
| EMITTER            | FREQ               | PRI        | TRANS. POWER | ANT. GAIN              | BEAM WIDTH | EFF. POWER | ANT. SCAN     |
| 1                  | 9.0400E 09         | 5.0000E-04 | 2.5000E 05   | 1.5000E 01             | 1.0000E 01 | 4.7631E 01 | CIR 2.00 SPR  |
| 2                  | 9.9000E 09         | 2.0000E-03 | 3.0000E 05   | 1.5000E 01             | 1.5000E 01 | 4.8192E 01 | CIR 5.00 SPR  |
| TARGET INFORMATION |                    |            |              |                        |            |            |               |
| 6                  | -200.0 48.0 0.0    | 0.0        | 20.          |                        |            |            |               |
| EMITTER            | FREQ               | PRI        | TRANS. POWER | ANT. GAIN              | BEAM WIDTH | EFF. POWER | ANT. SCAN     |
| 1                  | 9.2750E 09         | 1.0000E-03 | 1.5000E 04   | 2.4000E 01             | 4.0000E 00 | 5.1116E 01 | CIR 0.36 SPR  |
| 2                  | 9.5000E 09         | 4.0000E-03 | 5.0000E 04   | 3.0000E 01             | 5.0000E-01 | 5.5769E 01 | CIR 10.00 SPR |



DATA

| BEARING | PULSE | FREQ | BIN | DBLE | PRECISION | TIME | PW | TOA  |
|---------|-------|------|-----|------|-----------|------|----|------|
| 280     | 1     | 32   | 2   | 0    | 7381      | 1    | 10 | 8999 |
| 281     | 2     | 41   | 3   | 0    | 7421      | 1    | 20 | 9064 |
| 282     | 3     | 16   | 4   | 0    | 7446      | 1    | 10 | 9080 |
| 283     | 4     | 48   | 4   | 0    | 7454      | 1    | 10 | 9105 |
| 284     | 4     | 26   | 4   | 0    | 7464      | 1    | 15 | 9125 |
| 285     | 6     | 16   | 6   | 0    | 7470      | 1    | 20 | 9158 |
| 286     | 7     | 63   | 6   | 0    | 7482      | 1    | 25 | 9166 |
| 287     | 8     | 42   | 6   | 0    | 7486      | 1    | 18 | 9205 |
| 288     | 9     | 44   | 4   | 0    | 7503      | 1    | 10 | 9221 |
| 289     | 11    | 52   | 4   | 0    | 7507      | 1    | 5  | 9229 |
| 290     | 12    | 39   | 4   | 0    | 7537      | 1    | 4  | 9334 |
| 291     | 13    | 52   | 5   | 0    | 7547      | 1    | 10 | 9351 |
| 292     | 14    | 52   | 5   | 0    | 7550      | 1    | 5  | 9355 |
| 293     | 15    | 67   | 5   | 0    | 7556      | 1    | 17 | 9405 |
| 294     | 16    | 18   | 9   | 0    | 7558      | 1    | 50 | 9417 |
| 295     | 17    | 93   | 3   | 0    | 7576      | 1    | 25 | 9436 |
| 296     | 18    | 33   | 3   | 0    | 7579      | 1    | 10 | 9443 |
| 297     | 19    | 29   | 2   | 0    | 7584      | 1    | 15 | 9446 |
| 298     | 20    | 92   | 4   | 0    | 7588      | 1    | 20 | 9453 |
| 299     | 21    | 34   | 5   | 0    | 7593      | 1    | 5  | 9468 |
| 300     | 22    | 66   | 6   | 0    | 7595      | 1    | 10 | 9483 |
| 301     | 23    | 47   | 3   | 0    | 7596      | 1    | 15 | 9495 |
| 302     | 24    | 56   | 4   | 0    | 7600      | 1    | 20 | 9504 |
| 303     | 25    | 64   | 4   | 0    | 7604      | 1    | 5  | 9511 |
| 304     | 26    | 45   | 6   | 0    | 7610      | 1    | 1  | 9514 |
| 305     | 27    | 39   | 3   | 0    | 7615      | 1    | 20 | 9515 |
| 306     | 28    | 99   | 9   | 0    | 7621      | 1    | 4  | 9517 |
| 307     | 29    | 16   | 2   | 0    | 7628      | 1    | 1  | 9519 |
| 308     | 30    | 33   | 3   | 0    | 7632      | 1    | 10 | 9534 |
| 309     | 31    | 25   | 2   | 0    | 7635      | 1    | 15 | 9544 |
| 310     | 32    | 42   | 4   | 0    | 7636      | 1    | 10 | 9554 |
| 311     | 33    | 64   | 6   | 0    | 7644      | 1    | 5  | 9555 |
| 312     | 34    | 49   | 4   | 0    | 7644      | 1    | 1  | 9557 |
| 313     | 35    | 30   | 3   | 0    | 7648      | 1    | 10 | 9559 |
| 314     | 36    | 54   | 5   | 0    | 7654      | 1    | 5  | 9561 |
| 315     | 37    | 25   | 2   | 0    | 7663      | 1    | 1  | 9561 |
| 316     | 38    | 25   | 2   | 0    | 7663      | 1    | 1  | 9564 |
| 317     | 39    | 40   | 4   | 0    | 7670      | 1    | 1  | 9564 |
| 318     | 40    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 319     | 41    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 320     | 42    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 321     | 43    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 322     | 44    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 323     | 45    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 324     | 46    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 325     | 47    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 326     | 48    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 327     | 49    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 328     | 50    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 329     | 51    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 330     | 52    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 331     | 53    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 332     | 54    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 333     | 55    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 334     | 56    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 335     | 57    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 336     | 58    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 337     | 59    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 338     | 60    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 339     | 61    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 340     | 62    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 341     | 63    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 342     | 64    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 343     | 65    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 344     | 66    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 345     | 67    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 346     | 68    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 347     | 69    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 348     | 70    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 349     | 71    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 350     | 72    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 351     | 73    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 352     | 74    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 353     | 75    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 354     | 76    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 355     | 77    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 356     | 78    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 357     | 79    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 358     | 80    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 359     | 81    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 360     | 82    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 361     | 83    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 362     | 84    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 363     | 85    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 364     | 86    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 365     | 87    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 366     | 88    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 367     | 89    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 368     | 90    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 369     | 91    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 370     | 92    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 371     | 93    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 372     | 94    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 373     | 95    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 374     | 96    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 375     | 97    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 376     | 98    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 377     | 99    | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |
| 378     | 100   | 44   | 4   | 0    | 7674      | 1    | 1  | 9564 |



## APPENDIX C

### PULSE TRAIN SEPARATOR SIMULATION PROGRAM PL/ATAC LISTING

Program variables are described as they are declared in the program.



SIMPLE PL/ATAC COMPILER VERSION OF FEBRUARY 1, 1974. CLOCK TIME = 9:51:33.79.

TODAY IS MARCH 18, 1975. CLOCK TIME = 22:57:29.51.

```
1 (/**** TUGGLES: $PUNCH ) 1
2 (****/) ) 2
3 ) 3
4 (/** PL/ATAC PULSE TRAIN SEPARATOR ) 4
5 ) 5
6 ) 6
7 ) 7
8 ) 8
9 ) 9
10 THIS VERSION OF PL/ATAC CONTAINS NO DOUBLE PRECISION CAPABILITY ) 10
11 NECESSITATING EMPLOYMENT OF THE 'CODE' COLLECTION HEAD WHERE ) 11
12 DOUBLE PRECISION CALCULATIONS ARE PERFORMED. ) 12
13 ) 13
14 ) 14
15 ) 15
16 PROGRAM IS DESIGNED TO SIMULATE A HARDWARE SIGNAL PREPROCESSING ) 16
17 SYSTEM WITH A CAPABILITY OF SEPARATING UP TO A MAXIMUM OF TEN ) 17
18 STABLE PULSE TRAINS FROM THE OUTPUT OF AN IFM RECEIVER. THE IFM ) 18
19 RECEIVER OPERATES IN CONJUNCTION WITH A HIGH GAIN SCANNING ) 19
20 ANTENNA. FOR SIMULATION PURPOSES, DIRECTION OF ARRIVAL (DOA) IS ) 20
21 ASSUMED TO BE OBTAINED FROM AN AZIMUTH RESOLVER GATED BY THE ) 21
22 PULSE TRAIN SEPARATOR (PTS) HARDWARE. THE OUTPUT PULSES FROM THE ) 22
23 IFM RECEIVER FED TO THE PTS ARE SIMULATED BY LOADING A SERIES ) 23
24 OF FIVE WORDS FOR EACH PULSE RECEIVED. ) 24
25 1. DOA - IN DEGREES LSB = 1 DEG ) 25
26 2. FREQUENCY - IN TERMS OF FREQUENCY BIN OF THE PULSE ) 26
27 3,4. TIME OF ARRIVAL (TOA) - DOUBLE PRECISION VALUE ) 27
28 ) 28
29 ) 29
30 5. PULSE WIDTH (PW) - LSB = .1 MICROSECOND ) 30
31 ) 31
32 UPON RECEIPT OF FIVE PULSES OF A STABLE TRAIN, THE PTS LOCKS ) 32
33 ON THE TRAIN AND CONTINUES TO TRACK THE TRAIN FOR ONE ANTENNA ) 33
34 BEAMWIDTH. THE PULSE REPETITION INTERVAL (PRI) IS COMPUTED AND ) 34
35 AZIMUTH CENTROIDING, BASED ON DOA AND TOA OF THE FIRST AND LAST ) 35
36 PULSE OF THE TRAIN, IS EMPLOYED TO COMPUTE THE BEARING OF THE ) 36
37 EMITTER ASSOCIATED WITH THE TRAIN. THE PARAMETERS ASSOCIATED ) 37
38 WITH THE TRAIN, OUTPUT TO A MAINFRAME COMPUTER FOR SIGNAL ) 38
39 CLASSIFICATION, CONSIST OF A SERIES OF SIX WORDS. ) 39
40 1. EMITTER BEARING ) 40
41 2. FREQUENCY ) 41
42 3,4. TIME FIRST SEEN ) 42
43 5. PULSE WIDTH ) 43
44 6. PULSE REPETITION INTERVAL ) 44
45 THIS OUTPUT IS SIMULATED BY STORING THE PARAMETERS IN A MEMORY ) 45
46 BUFFER. ) 46
47 RESIDUAL PULSES PASS THROUGH A SEPARATE CHANNEL OF THE PTS TO A ) 47
48 HIGH SPEED MINI COMPUTER WHERE THEY ARE PROCESSED TO DETERMINE ) 48
49 PRI, JITTER AND STAGGER CHARACTERISTICS IF THEY EXIST. THIS ) 49
50 ACTION IS SIMULATED BY COMPRESSING THE RESIDUAL DATA WITHIN THE ) 50
51 INPUT BUFFER TO BE PROCESSED BY AN ATAC ASSEMBLY LANGUAGE ) 51
52 PROGRAM WHICH MAY BE INCORPORATED USING THE CODE COLLECTION HEAD ) 52
53 ) 53
54 (/**** G L O B A L V A R I A B L E S ****/) ) 54
55 ) 55
56 (/** DESCRIPTION OF GLOBAL VARIABLES ) 56
57 ) 57
58 RAM VARIABLES ) 58
59 ) 59
60 BUF 4000 WORD INPUT BUFFER ) 60
61 NS NEW SIGNAL POINTER FILE. HOLDS POINTERS TO A MAXIMUM OF 5 ) 61
62 PULSES FOR EACH EMITTER UP TO 40 EMITTERS. ) 62
63 TMIN DOUBLE PRECISION TIME FIRST SEEN ) 63
64 NC PTS ACTIVE CHANNEL COUNTER. INITIAL VALUE DETERMINES ) 64
65 NUMBER OF CHANNELS OF PTS. MAX NO CHANNELS = 10-NC INIT ) 65
66 P1 TIME INTERVAL BETWEEN PULSES ) 66
67 PRI PULSE REPETITION INTERVAL ) 67
68 CHAN 240 WORD BUFFER IN WHICH STABLE PULSE TRAIN PARAMETERS ARE ) 68
69 STORED ) 69
70 CNDX INDEX TO CHAN ) 70
71 CINDX INDEX TO CHAN ) 71
72 DP DATA POINTER ) 72
73 SSF STABLE SIGNAL FOUND ) 73
74 NSSF NON-STABLE SIGNAL FOUND ) 74
75 ) 75
76 ) 76
77 ) 77
```



```

79 ( ) 79
80 ( ) 80
81 ( ) 81
82 ( ) 82
83 ( ) 83
84 ( ) 84
85 ( ) 85
86 ( ) 86
87 ( ) 87
88 ( ) 88
89 ( ) 89
90 ( ) 90
91 ( ) 91
92 ( ) 92
93 ( ) 93
94 ( ) 94
95 ( ) 95
96 ( ) 96
97 ( ) 97
98 ( ) 98
99 ( ) 99
100 ( ) 100
101 ( ) 101
102 ( ) 102
103 ( ) 103
104 ( ) 104
105 ( ) 105
106 ( ) 106
107 ( ) 107
108 ( ) 108
109 ( ) 109
110 ( ) 110
111 ( ) 111
112 ( ) 112
113 ( ) 113
114 ( ) 114
115 ( ) 115
116 ( ) 116
117 ( ) 117
118 ( ) 118
119 ( ) 119
120 ( ) 120
121 ( ) 121
122 ( ) 122
123 ( ) 123
124 ( ) 124
125 ( ) 125
126 ( ) 126
127 ( ) 127
128 ( ) 128
129 ( ) 129
130 ( ) 130
131 ( ) 131
132 ( ) 132
133 ( ) 133
134 ( ) 134
135 ( ) 135
136 ( ) 136
137 ( ) 137
138 ( ) 138
139 ( ) 139
140 ( ) 140
141 ( ) 141
142 ( ) 142
143 ( ) 143
144 ( ) 144
145 ( ) 145
146 ( ) 146
147 ( ) 147
148 ( ) 148
149 ( ) 149
150 ( ) 150
151 ( ) 151
152 ( ) 152
153 ( ) 153
154 ( ) 154
155 ( ) 155
156 ( ) 156
157 ( ) 157
158 ( ) 158
159 ( ) 159
160 ( ) 160
161 ( ) 161
162 ( ) 162
163 ( ) 163
164 ( ) 164
165 ( ) 165
166 ( ) 166
167 ( ) 167
168 ( ) 168
169 ( ) 169
170 ( ) 170
171 ( ) 171
172 ( ) 172
173 ( ) 173
174 ( ) 174
175 ( ) 175
176 ( ) 176
177 ( ) 177
178 ( ) 178
179 ( ) 179
180 ( ) 180
181 ( ) 181
182 ( ) 182

(
REGISTER VARIABLES
)
(
1,J  NS FILE INDEXES
BP  BUFFER POINTER
)
(
"/
)
(
DECLARE BUF(3999)RAM
DECLARE NS(199)RAM
DECLARE TMIN(75)RAM
DECLARE (NC,PI(5),PRI(1),CNDX)RAM
DECLARE CHAN(120)RAM
DECLARE NSSF(39)RAM
DECLARE SSF(39)RAM
DECLARE CINDX(39)RAM
DECLARE (BP,1,J)REGISTER
DECLARE DP RAM
)
(
DEFINE TRUETOOL=(FALSE,TRUE)
)
(
LOCAL PROCEDURE PRICOMP
)
(
"/  PRICOMP DETERMINES THE STABILITY OF A PULSE TRAIN.  IF STABLE
THE PRI IS COMPUTED AND THE TRAIN IS FLAGGED AS A STABLE TRAIN.
ALL PARAMETERS EXCEPT OF ARE LOADED INTO THE CHANNEL BUFFER.
IF THE TRAIN IS NON STABLE IT IS FLAGGED AS SUCH.
)
(
DESCRIPTION OF VARIABLES
)
(
K,KK  INDEXES FOR CALCULATIONS
MP  MISSING PULSE COUNTER
PRITEM  CALCULATION VALUE OF PRI
)
(
"/
)
(
DECLARE (K,MP,KK,KJ,PRITEM)RAM
K=2
KK=KJ=1
MP=0
)
(
"/  TEST PULSE INTERVALS TO DETERMINE STABILITY OF TRAIN. A PULSE
TRAIN IS CONSIDERED STABLE IF NO MORE THAN TWO MISSING PULSES
ARE ENCOUNTERED IN INITIAL PULSES AND VARIATIONS IN TIME
INTERVAL BETWEEN PULSES DOES NOT EXCEED +- 51.2 MICROSECONDS.  "/
)
(
LOOP1:
DO
TEST@MP>2
IFSO: LEAVE LOOP1
END
TEST @PI(.@K-1.)/@KK IN (@PI(.@K.)/@KJ-2) TO (@PI(.@K.)/@KJ+2)
ABOVE: DO
MP=@MP+1
KK=@KK+1
REPEAT LOOP1
END
BELOW: DO
MP=@MP+1
KJ=@KJ+1
REPEAT LOOP1
END
END
K=@K+1
KK=@KJ
KJ=1
TEST @K>4
IFNOT: REPEAT LOOP1
END
)
(
"/  COMPUTE PRI AND TEST FOR ROUND OFF ERROR.  IF ERROR EXISTS
ROUND UP BY 1.
)
(
"/
)
(
K=1
PRI=0
ITERATE 4 TIMES
PRI=@PRI+@PI(.@K.)
K=@K+1
END
PRITEM=@PRI
PRI=@PRI/(4+@MP)
TEST @PRITEM MOD(4+@MP)>(4+@MP)/2
IFSO: PRI=@PRI+1
END
)
(
"/  LOAD PARAMETERS INTO CHANNEL BUFFER  "/
)
(
DP=@NS(.@1.)
CNDX=@CNDX+@
CINDX(.@1/5.)=@CNDX
CHAN(.@CNDX+1.)=@BUF(.@DP+1.)
CHAN(.@CNDX+2.)=@TMIN(.@1/5*2.)
CHAN(.@CNDX+3.)=@TMIN(.@1/5*2+1.)
CHAN(.@CNDX+4.)=@BUF(.@DP+.)
CHAN(.@CNDX+5.)=@PRI
)
(
"/  FLAG PULSE TRAIN AS STABLE AND INCREMENT CHANNEL COUNTER  "/
)
(
SSF(.@1/5.)=TRUE
NC=@NC+1
RETURN
)
(
END LOOP1
)

```



|     |                                                                    |     |         |
|-----|--------------------------------------------------------------------|-----|---------|
| 183 | (                                                                  | 183 | PRICOMP |
| 184 | /" TRAIN FAILED STABILITY TEST. FLAG TRAIN AS NON STABLE "/        | 184 | PRICOMP |
| 185 | (                                                                  | 185 | PRICOMP |
| 186 | NSSF(.@1/5).)=TRUE                                                 | 186 | PRICOMP |
| 187 | END PRICOMP                                                        | 187 | PRICOMP |
| 188 | (                                                                  | 188 |         |
| 189 |                                                                    | 189 |         |
| 190 | PROCEDURE MAIN                                                     | 190 |         |
| 191 | (                                                                  | 191 | MAIN    |
| 192 | /" DESCRIPTION OF VARIABLES                                        | 192 | MAIN    |
| 193 | (                                                                  | 193 | MAIN    |
| 194 | BRNGF RCVR ANTENNA BEARING AT TOA OF FIRST PULSE OF A SIGNAL       | 194 | MAIN    |
| 195 | BRNGL RCVR ANTENNA BEARING AT TOA OF LAST PULSE OF A TRAIN         | 195 | MAIN    |
| 196 | CUN1 DOUBLE PRECISION TIME CALCULATION VARIABLE                    | 196 | MAIN    |
| 197 | CUN2 DOUBLE PRECISION TIME CALCULATION VARIABLE                    | 197 | MAIN    |
| 198 | II,JJ INDICES                                                      | 198 | MAIN    |
| 199 | SCANTM RCVR ANTENNA SCAN TIME FOR ONE BEAMWIDTH                    | 199 | MAIN    |
| 200 | TCAL LOWER HALF DOUBLE PRECISION TIME OF ARRIVAL                   | 200 | MAIN    |
| 201 | TOAU UPPER HALF DOUBLE PRECISION TIME OF ARRIVAL                   | 201 | MAIN    |
| 202 | TMAX EXPECTED TIME OF LAST PULSE INTERCEPT                         | 202 | MAIN    |
| 203 | TMIN DOUBLE PRECISION TIME OF ARRIVAL OF FIRST PULSE               | 203 | MAIN    |
| 204 | ITEM DOUBLE PRECISION TIME CALCULATION VARIABLE                    | 204 | MAIN    |
| 205 | TTEST DOUBLE PRECISION TIME CALCULATION VARIABLE                   | 205 | MAIN    |
| 206 | /"                                                                 | 206 | MAIN    |
| 207 | (                                                                  | 207 | MAIN    |
| 208 | DECLARE(BRNGF,BRNGL(39),SCANTM(1),II,JJ,STR)AM                     | 208 | MAIN    |
| 209 | DECLARE(TCAL,TOAU,TMAX(79),TTEST(1),ITEM(1))AM                     | 209 | MAIN    |
| 210 | DECLARE(CUN1(1),CUN2(1))AM                                         | 210 | MAIN    |
| 211 | (                                                                  | 211 | MAIN    |
| 212 | /" INITIALIZE VARIABLES "/                                         | 212 | MAIN    |
| 213 | (                                                                  | 213 | MAIN    |
| 214 | CNDX=-6                                                            | 214 | MAIN    |
| 215 | SCANTM=1120                                                        | 215 | MAIN    |
| 216 | I=0                                                                | 216 | MAIN    |
| 217 | BP=0                                                               | 217 | MAIN    |
| 218 | NC=1                                                               | 218 | MAIN    |
| 219 | (                                                                  | 219 | MAIN    |
| 220 | /" SET NEW SIGNAL PCINTER FILE TO NEGATIVE VALUES "/               | 220 | MAIN    |
| 221 | (                                                                  | 221 | MAIN    |
| 222 | ITERATE 200 TIMES                                                  | 222 | MAIN    |
| 223 | NS(.@1.)=-1                                                        | 223 | MAIN    |
| 224 | I=@I+1                                                             | 224 | MAIN    |
| 225 | END                                                                | 225 | MAIN    |
| 226 | I=0                                                                | 226 | MAIN    |
| 227 | NS(.@1.)=0                                                         | 227 | MAIN    |
| 228 | (                                                                  | 228 | MAIN    |
| 229 | /" LOAD INPUT BUFFER FROM CARDS "/                                 | 229 | MAIN    |
| 230 | (                                                                  | 230 | MAIN    |
| 231 | READ_DEC(BUF,BUF+3999)                                             | 231 | MAIN    |
| 232 | (                                                                  | 232 | MAIN    |
| 233 | MAINLP:                                                            | 233 | MAIN    |
| 234 | % DO                                                               | 234 | MAIN    |
| 235 | (                                                                  | 235 | MAIN    |
| 236 | /" LOAD UP TOA OF FIRST PULSE OF SIGNAL FROM INPUT BUFFER AND      | 236 | MAIN    |
| 237 | COMPUTE EXPECTED TIME OF ARRIVAL OF LAST PULSE BASED ON RCVR       | 237 | MAIN    |
| 238 | ANTENNA SCAN TIME FOR ONE ANTENNA BEAMWIDTH "/                     | 238 | MAIN    |
| 239 | (                                                                  | 239 | MAIN    |
| 240 | TCAL=@BUF(.@BP+2.)                                                 | 240 | MAIN    |
| 241 | TOAU=@BUF(.@BP+3.)                                                 | 241 | MAIN    |
| 242 | CCDE                                                               | 242 | MAIN    |
| 243 | LDRM D,14,TCAL,2                                                   | 243 | MAIN    |
| 244 | DADD D,14,SCANTM                                                   | 244 | MAIN    |
| 245 | STRM D,14,ITEM,2                                                   | 245 | MAIN    |
| 246 | END                                                                | 246 | MAIN    |
| 247 | STR=@I/5*2                                                         | 247 | MAIN    |
| 248 | TMAX(.@STR.)=@ITEM                                                 | 248 | MAIN    |
| 249 | TMAX(.@STR+1.)=@ITEM(.1.)                                          | 249 | MAIN    |
| 250 | TMIN(.@STR.)=@TOAU                                                 | 250 | MAIN    |
| 251 | TMIN(.@STR+1.)=@TOAU                                               | 251 | MAIN    |
| 252 | COMPUTE:                                                           | 252 | MAIN    |
| 253 | % DO                                                               | 253 | MAIN    |
| 254 | (                                                                  | 254 | MAIN    |
| 255 | /" INCREMENT BUFFER POINTER AND CHECK AT BUFFER LIMIT. IFSO, GO    | 255 | MAIN    |
| 256 | COMPRESS RESIDUAL PULSES WITHIN INPUT BUFFER FOR FUTHER PROCESSING | 256 | MAIN    |
| 257 | /"                                                                 | 257 | MAIN    |
| 258 | (                                                                  | 258 | MAIN    |
| 259 | BP=@BP+5                                                           | 259 | MAIN    |
| 260 | TEST @BP>3999                                                      | 260 | MAIN    |
| 261 | IFSO: LEAVE MAINLP                                                 | 261 | MAIN    |
| 262 | END                                                                | 262 | MAIN    |
| 263 | (                                                                  | 263 | MAIN    |
| 264 | /" IF NO SIGNAL IN BUFFER INDEXED BY BUFFER POINTER INCREMENT      | 264 | MAIN    |
| 265 | PCINTER AND REPEAT COMPUTE "/                                      | 265 | MAIN    |
| 266 | (                                                                  | 266 | MAIN    |
| 267 | TEST @BUF(.@BP.)                                                   | 267 | MAIN    |
| 268 | ZERO: REPEAT COMPUTE                                               | 268 | MAIN    |
| 269 | END                                                                | 269 | MAIN    |
| 270 | I=0                                                                | 270 | MAIN    |
| 271 | J=0                                                                | 271 | MAIN    |
| 272 | SORT:                                                              | 272 | MAIN    |
| 273 | % DO                                                               | 273 | MAIN    |
| 274 | (                                                                  | 274 | MAIN    |
| 275 | /" IF AT END OF NEW SIGNAL POINTER FILE, LEAVE SORT "/             | 275 | MAIN    |
| 276 | (                                                                  | 276 | MAIN    |
| 277 | TEST @I>195                                                        | 277 | MAIN    |
| 278 | IFSO: LEAVE SORT                                                   | 278 | MAIN    |
| 279 | END                                                                | 279 | MAIN    |
| 280 | (                                                                  | 280 | MAIN    |
| 281 | /" IF NEW SIGNAL POINTER FILE INDEXED BY I IS EMPTY INDEX AND      | 281 | MAIN    |
| 282 | CHECK NEXT LOCATION "/                                             | 282 | MAIN    |
| 283 | (                                                                  | 283 | MAIN    |
| 284 | TEST @NS(.@1.)                                                     | 284 | MAIN    |
| 285 | NEG: DO                                                            | 285 | MAIN    |
| 286 | I=@I+5                                                             | 286 | MAIN    |
| 287 | REPEAT SORT                                                        | 287 | MAIN    |



```

288 ( END
289 ( END
290 (
291 (/" COMPARE TOA OF BUFFER SIGNAL WITH TMAX OF SIGNAL FOUND IN NEW
292 ( SIGNAL POINTER FILE. IF TOA OF BUFFER SIGNAL IS GREATER THAN
293 ( OR EQUAL TO TMAX, TEST FOR STABLE SIGNAL IN NS POINTER FILE.
294 ( IF SIGNAL HAS BEEN CLASSIFIED STABLE, RESET ACTIVE CHANNEL
295 ( COUNTER AND SSF FLAG, ZERO FIRST FIVE PULSES FROM THIS SIGNAL
296 ( IN INPUT BUFFER AND COMPUTE EMITTER BEARING USING FIRST AND
297 ( DOA OF SIGNAL "/
298 (
299 ( CON1=@BUF(.@BP+2).)
300 ( CLN1(.1.)=@BUF(.@BP+3).)
301 ( CON2=@TMAX(.@I/5*2).)
302 ( CON2(.1.)=@TMAX(.@I/5*2+1).)
303 ( CODE
304 ( LDRM D,14,CON1,2
305 ( LDRM D,7,CON2,2
306 ( DSUB R,7,14
307 ( STRM D,7,TTEST,2
308 (
309 ( END
310 ( TEST @TTEST
311 ( NEG @TTEST: DO
312 ( TEST @SSF(.@I/5.)
313 ( POS: DO
314 ( NC=@NC-1
315 ( SSF(.@I/5.)=FALSE
316 ( BRNGF=@BUF(.@NS(.@I.)).)
317 ( II=0
318 ( ITERATE 5 TIMES
319 ( JJ=0
320 ( ITERATE 5 TIMES
321 ( EUF(.@NS(.@I+@II.)+@JJ.)=0
322 ( JJ=@JJ+1
323 ( END
324 ( II=@II+1
325 ( END
326 ( CHAN(.@CINDX(.@I/5.))=(@BRNGF+@BRNGL(.@I/5.))/2
327 ( END
328 (
329 ( END
330 (/" IF NS POINTER FILE SIGNAL HAS BEEN CLASSIFIED AS NON STABLE
331 ( RESET NSSF FLAG "/
332 (
333 ( NSSF(.@I/5.)=FALSE
334 (
335 (/" CLEAR THE SIGNAL FROM NS POINTER FILE "/
336 (
337 ( II=0
338 ( ITERATE 5 TIMES
339 ( NS(.@I+@II.)=-1
340 ( II=@II+1
341 ( END
342 ( I=@I+5
343 ( REPEAT SORT
344 ( END
345 ( END
346 ( TEST @BUF(.@BP+1.)=@BUF(.@NS(.@I.)+1.)
347 ( IFNOT: DO
348 ( I=@I+5
349 ( REPEAT SCKT
350 ( END
351 ( END
352 (/" TEST FOR MATCH BETWEEN BUFFER SIGNAL FREQ AND PW AND NS SIGNAL
353 ( POINTER FILE FREQ AND PW. IF BOTH DO NOT MATCH REPEAT SORT.
354 ( IF MATCH OCCURS AND SIGNAL HAS BEEN CLASSIFIED AS STABLE,
355 ( ZERO BUFFER SIGNAL AND REPEAT COMPUTE. IF SIGNAL HAS BEEN
356 ( CLASSIFIED AS NON STABLE, REPEAT COMPUTE. "/
357 (
358 ( TEST @BUF(.@BP+4.)=@BUF(.@NS(.@I.)+4.)
359 ( IFNOT: DO
360 ( I=@I+5
361 ( REPEAT SORT
362 ( END
363 ( END
364 ( TEST @SSF(.@I/5.)
365 ( NONZERO: DO
366 ( BRNGL(.@I/5.)=@BUF(.@BP.)
367 ( II=0
368 ( ITERATE 5 TIMES
369 ( BUF(.@BP+@II.)=0
370 ( II=@II+1
371 ( END
372 ( REPEAT COMPUTE
373 ( END
374 ( END
375 ( TEST @NSSF(.@I/5.)=TRUE
376 ( IFSO: REPEAT COMPUTE
377 ( END
378 (
379 (/" IF SIGNAL HAS NOT BEEN CLASSIFIED, STORE BUFFER POINTER IN
380 ( NS POINTER FILE. IF THIS IS THE FIFTH PULSE ASSOCIATED WITH
381 ( THIS TRAIN, COMPUTE TIME INTERVALS BETWEEN PULSES AND CALL
382 ( FOR PK1 COMPUTATION. "/
383 (
384 ( ITERATE 3 TIMES
385 ( J=@J+1
386 ( TEST @NS(.@I+@J.)
387 ( NEG: DO
388 ( NS(.@I+@J.)=@BP
389 ( REPEAT COMPUTE
390 ( END
391 ( END
392 ( END

```

```

288 MAIN
289 MAIN
290 MAIN
291 MAIN
292 MAIN
293 MAIN
294 MAIN
295 MAIN
296 MAIN
297 MAIN
298 MAIN
299 MAIN
300 MAIN
301 MAIN
302 MAIN
303 MAIN
304 MAIN
305 MAIN
306 MAIN
307 MAIN
308 MAIN
309 MAIN
310 MAIN
311 MAIN
312 MAIN
313 MAIN
314 MAIN
315 MAIN
316 MAIN
317 MAIN
318 MAIN
319 MAIN
320 MAIN
321 MAIN
322 MAIN
323 MAIN
324 MAIN
325 MAIN
326 MAIN
327 MAIN
328 MAIN
329 MAIN
330 MAIN
331 MAIN
332 MAIN
333 MAIN
334 MAIN
335 MAIN
336 MAIN
337 MAIN
338 MAIN
339 MAIN
340 MAIN
341 MAIN
342 MAIN
343 MAIN
344 MAIN
345 MAIN
346 MAIN
347 MAIN
348 MAIN
349 MAIN
350 MAIN
351 MAIN
352 MAIN
353 MAIN
354 MAIN
355 MAIN
356 MAIN
357 MAIN
358 MAIN
359 MAIN
360 MAIN
361 MAIN
362 MAIN
363 MAIN
364 MAIN
365 MAIN
366 MAIN
367 MAIN
368 MAIN
369 MAIN
370 MAIN
371 MAIN
372 MAIN
373 MAIN
374 MAIN
375 MAIN
376 MAIN
377 MAIN
378 MAIN
379 MAIN
380 MAIN
381 MAIN
382 MAIN
383 MAIN
384 MAIN
385 MAIN
386 MAIN
387 MAIN
388 MAIN
389 MAIN
390 MAIN
391 MAIN
392 MAIN

```



|       |                                                              |            |
|-------|--------------------------------------------------------------|------------|
| 393 ( | NS(.@I+4.)=@BP                                               | ) 393 MAIN |
| 394 ( | TEST @NC==10                                                 | ) 394 MAIN |
| 395 ( | IFSO: DD                                                     | ) 395 MAIN |
| 396 ( | NSSF(.@I/5.)=TRUE                                            | ) 396 MAIN |
| 397 ( | REPEAT COMPUTE                                               | ) 397 MAIN |
| 398 ( | END                                                          | ) 398 MAIN |
| 399 ( | END                                                          | ) 399 MAIN |
| 400 ( | J=1                                                          | ) 400 MAIN |
| 401 ( | ITERATE 4 TIMES                                              | ) 401 MAIN |
| 402 ( | PI(.@J.)=@BUF(.(@NS(.@I+@J.)+2.)-@BUF(.(@NS(.@I+@J-1.)+2.)   | ) 402 MAIN |
| 403 ( | TEST @PI(.@J.)                                               | ) 403 MAIN |
| 404 ( | NEG: PI(.@J.)=@PI(.@J.)+32767                                | ) 404 MAIN |
| 405 ( | END                                                          | ) 405 MAIN |
| 406 ( | J=@J+1                                                       | ) 406 MAIN |
| 407 ( | END                                                          | ) 407 MAIN |
| 408 ( | CALL PRICOMP()                                               | ) 408 MAIN |
| 409 ( | REPEAT COMPUTE                                               | ) 409 MAIN |
| 410 ( | END SCRT                                                     | ) 410 MAIN |
| 411 ( | I=0                                                          | ) 411 MAIN |
| 412 ( | CYCLE                                                        | ) 412 MAIN |
| 413 ( | TEST @NS(.@I.)                                               | ) 413 MAIN |
| 414 ( | NEG: DD                                                      | ) 414 MAIN |
| 415 ( | NS(.@I.)=@BP                                                 | ) 415 MAIN |
| 416 ( | REPEAT MAINLP                                                | ) 416 MAIN |
| 417 ( | END                                                          | ) 417 MAIN |
| 418 ( | END                                                          | ) 418 MAIN |
| 419 ( | TEST(I=@I+5)>195                                             | ) 419 MAIN |
| 420 ( | IFSO: DD                                                     | ) 420 MAIN |
| 421 ( | DUMP_DEC(NS,NS+195)                                          | ) 421 MAIN |
| 422 ( | RETURN                                                       | ) 422 MAIN |
| 423 ( | END                                                          | ) 423 MAIN |
| 424 ( | END                                                          | ) 424 MAIN |
| 425 ( | END                                                          | ) 425 MAIN |
| 426 ( | END COMPUTE                                                  | ) 426 MAIN |
| 427 ( | END MAINLP                                                   | ) 427 MAIN |
| 428 ( |                                                              | ) 428 MAIN |
| 429 ( | /" ROUTINE BELCW COMPRESSES RESIDUAL PULSES WITHIN THE INPUT | ) 429 MAIN |
| 430 ( | BUFFER. AN ATAC ASSEMBLY LANGUAGE PROGRAM TO PROCESS THIS    | ) 430 MAIN |
| 431 ( | RESIDUAL MAY BE INCORPORATED BETWEEN THE END AND END MAIN    | ) 431 MAIN |
| 432 ( | STATEMENTS UTILIZING THE CODE COLLECTION HEAD. "/            | ) 432 MAIN |
| 433 ( |                                                              | ) 433 MAIN |
| 434 ( | BP=0                                                         | ) 434 MAIN |
| 435 ( | ZEROCK:                                                      | ) 435 MAIN |
| 436 ( | CYCLE                                                        | ) 436 MAIN |
| 437 ( | TEST @BUF(.@BP+1.)                                           | ) 437 MAIN |
| 438 ( | ZERD: DD                                                     | ) 438 MAIN |
| 439 ( | DP=@BP                                                       | ) 439 MAIN |
| 440 ( | LEAVE ZEROCK                                                 | ) 440 MAIN |
| 441 ( | END                                                          | ) 441 MAIN |
| 442 ( | END                                                          | ) 442 MAIN |
| 443 ( | BP=@BP+5                                                     | ) 443 MAIN |
| 444 ( | END ZEROCK                                                   | ) 444 MAIN |
| 445 ( | CYCLE                                                        | ) 445 MAIN |
| 446 ( | BP=@BP+5                                                     | ) 446 MAIN |
| 447 ( | TEST @BP>3995                                                | ) 447 MAIN |
| 448 ( | IFSO: DD                                                     | ) 448 MAIN |
| 449 ( | RETURN                                                       | ) 449 MAIN |
| 450 ( | END                                                          | ) 450 MAIN |
| 451 ( | END                                                          | ) 451 MAIN |
| 452 ( | TEST @BUF(.@BP+1.)==0                                        | ) 452 MAIN |
| 453 ( | IFNOT: DD                                                    | ) 453 MAIN |
| 454 ( | I=0                                                          | ) 454 MAIN |
| 455 ( | ITERATE 5 TIMES                                              | ) 455 MAIN |
| 456 ( | BUF(.@DP+@I.)=@BUF(.@BP+@I.)                                 | ) 456 MAIN |
| 457 ( | BUF(.@BP+@I.)=0                                              | ) 457 MAIN |
| 458 ( | I=@I+1                                                       | ) 458 MAIN |
| 459 ( | END                                                          | ) 459 MAIN |
| 460 ( | DP=@DP+5                                                     | ) 460 MAIN |
| 461 ( | END                                                          | ) 461 MAIN |
| 462 ( | END                                                          | ) 462 MAIN |
| 463 ( | END                                                          | ) 463 MAIN |
| 464 ( | END MAIN                                                     | ) 464 MAIN |
| 465 ( |                                                              | ) 465      |
| 466 ( |                                                              | ) 466      |
| 467 ( | /"**** LINK MODULE FOR EXECUTING PL/ATAC PROGRAM.            | ) 467      |
| 468 ( | \$EXECUTIVE                                                  | ) 468      |
| 469 ( | ****"/                                                       | ) 469      |
| 470 ( | MODULE LINK                                                  | ) 470      |
| 471 ( | DECLARE PROCSTK(64)RAM                                       | ) 471 LINK |
| 472 ( | RD = PROCSTK                                                 | ) 472 LINK |
| 473 ( | CALL MAIN()                                                  | ) 473 LINK |
| 474 ( | END LINK                                                     | ) 474 LINK |
| 475 ( | EOF EOF EOF EOF EOF EOF EOF EOF                              | ) 475      |



## APPENDIX D

PULSE TRAIN SEPARATOR SIMULATION PROGRAM  
ASSEMBLY LANGUAGE LISTING AND OUTPUT



LOC OBJECT CODE CARD IMAGE

CARDNUM

```
1 ****"/
2 /* PL/ATAC PULSE TRAIN SEPARATOR
3
4
5 ***** N U T E *****
6
7 THIS VERSION OF PL/ATAC CONTAINS NO DOUBLE PRECISION CAPABILITY
8 NECESSITATING EMPLOYMENT OF THE CODE COLLECTION HEAD WHERE
9 DOUBLE PRECISION CALCULATIONS ARE PERFORMED.
10
11 *****
12
13 PROGRAM IS DESIGNED TO SIMULATE A HARDWARE SIGNAL PREPROCESSING
14 SYSTEM WITH A CAPABILITY OF SEPARATING UP TO A MAXIMUM OF TEN
15 STABLE PULSE TRAINS FROM THE OUTPUT OF AN IFM RECEIVER. THE IFM
16 RECEIVER OPERATES IN CONJUNCTION WITH A HIGH GAIN SCANNING
17 ANTENNA. FOR SIMULATION PURPOSES, DIRECTION OF ARRIVAL (DOA) IS
18 ASSUMED TO BE OBTAINED FROM AN AZIMUTH RESOLVER GATED BY THE
19 PULSE TRAIN SEPARATOR (PTS) HARDWARE. THE OUTPUT PULSES FROM THE
20 IFM RECEIVER FED TO THE PTS ARE SIMULATED BY LOADING A SERIES
21 OF FIVE WORDS FOR EACH PULSE RECEIVED.
22
23 1. DOA - IN DEGREES LSB = 1 DEG
24 2. FREQUENCY - IN TERMS OF FREQUENCY BIN OF THE PULSE
25 3,4. TIME OF ARRIVAL (TOA) - DOUBLE PRECISION VALUE
26
27 5. PULSE WIDTH (PW) - LSB = .1 MICROSECOND
28
29 UPON RECEIPT OF FIVE PULSES OF A STABLE TRAIN, THE PTS LOCKS
30 ON THE TRAIN AND CONTINUES TO TRACK THE TRAIN FOR ONE ANTENNA
31 BEAMWIDTH. THE PULSE REPETITION INTERVAL (PRI) IS COMPUTED AND
32 AZIMUTH CENTROIDING, BASED ON DOA AND TOA OF THE FIRST AND LAST
33 PULSE OF THE TRAIN, IS EMPLOYED TO COMPUTE THE BEARING OF THE
34 EMITTER ASSOCIATED WITH THE TRAIN. THE PARAMETERS ASSOCIATED
35 WITH THE TRAIN, OUTPUT TO A MAINFRAME COMPUTER FOR SIGNAL
36 CLASSIFICATION, CONSIST OF A SERIES OF SIX WORDS.
37
38 1. EMITTER BEARING
39 2. FREQUENCY
40 3,4. TIME FIRST SEEN
41 5. PULSE WIDTH
42 6. PULSE REPETITION INTERVAL
43
44 THIS OUTPUT IS SIMULATED BY STORING THE PARAMETERS IN A MEMORY
45 BUFFER.
46 RESIDUAL PULSES PASS THROUGH A SEPARATE CHANNEL OF THE PTS TO A
47 HIGH SPEED MINI COMPUTER WHERE THEY ARE PROCESSED TO DETERMINE
48 PRI, JITTER AND STAGGER CHARACTERISTICS IF THEY EXIST. THIS
49 ACTION IS SIMULATED BY COMPRESSING THE RESIDUAL DATA WITHIN THE
50 INPUT BUFFER TO BE PROCESSED BY AN ATAC ASSEMBLY LANGUAGE
51 PROGRAM WHICH MAY BE INCORPORATED USING THE CODE COLLECTION HEAD
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

/\* \*\*\*\*\* GLOBAL VARIABLES \*\*\*\*\* \*/

/\* DESCRIPTION OF GLOBAL VARIABLES



| LOC  | OBJECT CODE | CARD IMAGE                                                 | CARDNUM |
|------|-------------|------------------------------------------------------------|---------|
|      |             | RAM VARIABLES                                              |         |
|      |             | 4000 WORD INPUT BUFFER                                     | 55      |
|      |             | NEW SIGNAL POINTER FILE. HOLDS POINTERS TO A MAXIMUM OF 5  | 56      |
|      |             | PULSES FOR EACH EMITTER UP TO 40 EMITTERS.                 | 57      |
|      |             | DOUBLE PRECISION TIME FIRST SEEN                           | 58      |
|      |             | PTS ACTIVE CHANNEL COUNTER. INITIAL VALUE DETERMINES       | 59      |
|      |             | NUMBER OF CHANNELS OF PTS. MAX NO CHANNELS = 10-NC INIT    | 60      |
|      |             | TIME INTERVAL BETWEEN PULSES                               | 61      |
|      |             | PULSE REPTITION INTERVAL                                   | 62      |
|      |             | 240 WORD BUFFER IN WHICH STABLE PULSE TRAIN PARAMETERS ARE | 63      |
|      |             | STORED                                                     | 64      |
|      |             | CNDX INDEX TO CHAN                                         | 65      |
|      |             | CINDX INDEX TO CHAN                                        | 66      |
|      |             | DATA POINTER                                               | 67      |
|      |             | STABLE SIGNAL FOUND                                        | 68      |
|      |             | NSSF NON-STABLE SIGNAL FOUND                               | 69      |
|      |             | REGISTER VARIABLES                                         | 70      |
|      |             | I,J NS FILE INDEXES                                        | 71      |
|      |             | BP BUFFER POINTER                                          | 72      |
|      |             | "/                                                         | 73      |
|      |             | DECLARE BUF(3999)RAM                                       | 74      |
|      |             | ORG RAM                                                    | 75      |
|      |             | DS 4000                                                    | 76      |
|      |             | DECLARE NS(199)RAM                                         | 77      |
|      |             | DS 200                                                     | 78      |
|      |             | DECLARE TMIN(79)RAM                                        | 79      |
|      |             | DS 80                                                      | 80      |
|      |             | DECLARE INC,PI(51,PRI(1),CNDX)RAM                          | 81      |
|      |             | NC DS 1                                                    | 82      |
|      |             | PI DS 5                                                    | 83      |
|      |             | PRI DS 2                                                   | 84      |
|      |             | CNDX DS 1                                                  | 85      |
|      |             | DECLARE CHAN(120)RAM                                       | 86      |
|      |             | CHAN DS 121                                                | 87      |
|      |             | DECLARE NSSF(39)RAM                                        | 88      |
|      |             | NSSF DS 40                                                 | 89      |
|      |             | DECLARE SSF(39) RAM                                        | 90      |
|      |             | SSF DS 40                                                  | 91      |
|      |             | DECLARE CINDX(39) RAM                                      | 92      |
|      |             | CINDX DS 40                                                | 93      |
|      |             | DECLARE (BP,I,J)REGISTER                                   | 94      |
|      |             | DP RAM                                                     | 95      |
|      |             | DP DS 1                                                    | 96      |
| 2000 |             |                                                            | 100     |
| 2FA0 |             |                                                            | 101     |
| 3068 |             |                                                            | 102     |
| 3088 |             |                                                            | 103     |
| 3089 |             |                                                            | 104     |
| 30BF |             |                                                            | 105     |
| 30C1 |             |                                                            | 106     |
| 30C2 |             |                                                            | 107     |
| 3138 |             |                                                            |         |
| 3163 |             |                                                            |         |
| 318B |             |                                                            |         |
| 3185 |             |                                                            |         |







| LJC  | OBJECT CODE | CARD IMAGE      | CARDNUM |
|------|-------------|-----------------|---------|
| 000E | C106 0012   | BRC I,6,a5      | 161     |
| 0010 | C107 00F1   | BRC I,7,a1      | 162     |
|      |             | END             | 163     |
|      |             | TEST            | 164     |
| 0012 | E202 31B4   | EQU \$          | 165     |
| 0014 | 6FF2        | EQU \$          | 166     |
| 0015 | E324 30B9   | LDR D,2,0+K     | 167     |
| 0017 | 08F3        | ADD IS,2,-1     | 168     |
| 0018 | 08F3 31B6   | LDR DX,4,0+PI,2 | 169     |
| 001A | E205        | SHD RA,3,16     | 170     |
| 001B | FC53        | LDR O,5,0+KK    | 171     |
| 001D | E205 31B4   | DIV R,3,5       | 172     |
| 001F | E357 30B9   | LDR D,5,0+K     | 173     |
| 0020 | D8F6        | LDR DX,7,0+PI,5 | 174     |
| 0022 | E208 31B7   | SHD RA,6,16     | 175     |
| 0023 | 6FE7        | LDR D,8,0+KJ    | 176     |
| 0024 | E208 31B4   | DIV R,6,8       | 177     |
| 0026 | E383 30B9   | ADD IS,7,-2     | 178     |
| 0028 | D8F2        | LDR D,8,0+K     | 179     |
| 0029 | E205 31B7   | LDR DX,3,0+PI,8 | 180     |
| 002B | FC52        | SHD RA,2,16     | 181     |
| 002C | 6023        | LDR D,5,0+KJ    | 182     |
| 002D | E035        | DIV R,2,5       | 183     |
| 002E | E036        | ADD IS,3,2      | 184     |
| 002F | E076        | LDR R,5,3       | 185     |
| 0030 | 8854        | LDR R,5,5       | 186     |
|      |             | SUB R,6,7       | 187     |
|      |             | CBL R,4,5       | 188     |
|      |             | ABOVE: 00       | 189     |
| 0031 | C106 0041   | EQU \$          | 190     |
| 0033 | E205 31B5   | BRC I,6,a9      | 191     |
| 0035 | 6015        | MP=AMP+1        | 192     |
| 0036 | 9C05 31B5   | D,5,0+MP        | 193     |
|      |             | ADD IS,5,1      | 194     |
| 0038 | E205 31B6   | STR D,5,0+MP    | 195     |
| 003A | 6015        | KK=KK+1         | 196     |
| 003B | 9C05 31B6   | LDR D,5,0+KK    | 197     |
| 003D | C107 0008   | ADD IS,5,1      | 198     |
|      |             | STR D,5,0+KK    | 199     |
|      |             | REPEAT LOOP1    | 200     |
|      |             | I,7,a0          | 201     |
|      |             | BRC END         | 202     |
|      |             | DO              | 203     |
| 003F | C107 004F   | DO I,7,a6       | 204     |
| 0041 | C105 004F   | BRC \$          | 205     |
| 0043 | E205 31B5   | EQU I,3,a10     | 206     |
| 0045 | 6015        | BRC MP=AMP+1    | 207     |
| 0046 | 9C05 31B5   | LDR D,5,0+MP    | 208     |
|      |             | ADD IS,5,1      | 209     |
|      |             | STR D,5,0+MP    | 210     |
| 0048 | E205 31B7   | KJ=AKJ+1        | 211     |
| 004A | 6015        | LDR O,5,0+KJ    | 212     |
|      |             | ADD IS,5,1      | 213     |



CARDNUM

| LOC  | OBJECT CODE | CARD IMAGE                                                | CARDNUM |
|------|-------------|-----------------------------------------------------------|---------|
| 004B | 9C05 31B7   | STR D,5,0+KJ                                              | 214     |
| 004D | C1C7 000B   | REPEAT LOOP1                                              | 215     |
|      |             | I,7,a0                                                    | 216     |
|      |             | END                                                       | 217     |
|      |             | END                                                       | 218     |
|      |             | EQU \$                                                    | 219     |
|      |             | EQU \$                                                    | 220     |
| 004F | E205 31B4   | K=@K+1                                                    | 221     |
| 0051 | 6015        | LDR D,5,0+K                                               | 222     |
| 0052 | 9C05 31B4   | ADD IS,5,1                                                | 223     |
|      |             | STR D,5,0+K                                               | 224     |
| 0054 | E205 31B7   | LDR D,5,0+KJ                                              | 225     |
| 0056 | 9C05 31B6   | STR D,5,0+KK                                              | 226     |
| 0058 | 4015        | LDR IS,5,1                                                | 227     |
| 0059 | 9C05 31B7   | STR D,5,0+KJ                                              | 228     |
| 005B | E205 31B4   | LDR D,5,0+K                                               | 229     |
| 005D | 2045        | IS,5,4                                                    | 230     |
|      |             | CMP REPEAT LOOP1                                          | 231     |
|      |             | IFNOT                                                     | 232     |
|      |             | EQU \$                                                    | 233     |
|      |             | BRC I,1,a14                                               | 234     |
|      |             | BRC I,7,a0                                                | 235     |
| 005E | C101 0062   | END                                                       | 236     |
| 0060 | C107 000B   | EQU \$                                                    | 237     |
|      |             | EQU \$                                                    | 238     |
|      |             | COMPUTE PRI AND TEST FOR ROUND OFF ERROR. IF ERROR EXISTS | 239     |
|      |             | ROUND UP BY 1.                                            | 240     |
|      |             | /"                                                        | 241     |
|      |             | "/                                                        | 242     |
|      |             | ."                                                        | 243     |
|      |             | ."                                                        | 244     |
|      |             | ."                                                        | 245     |
|      |             | ."                                                        | 246     |
|      |             | ."                                                        | 247     |
|      |             | ."                                                        | 248     |
|      |             | ."                                                        | 249     |
|      |             | ."                                                        | 250     |
|      |             | ."                                                        | 251     |
|      |             | ."                                                        | 252     |
|      |             | ."                                                        | 253     |
|      |             | ."                                                        | 254     |
|      |             | ."                                                        | 255     |
|      |             | ."                                                        | 256     |
|      |             | ."                                                        | 257     |
|      |             | ."                                                        | 258     |
|      |             | ."                                                        | 259     |
|      |             | ."                                                        | 260     |
|      |             | ."                                                        | 261     |
|      |             | ."                                                        | 262     |
|      |             | ."                                                        | 263     |
|      |             | ."                                                        | 264     |
|      |             | ."                                                        | 265     |
|      |             | ."                                                        | 266     |
| 0062 | 4015        | LDR IS,5,1                                                | 267     |
| 0063 | 9C05 31B4   | STR D,5,0+K                                               | 268     |
| 0065 | 9C0C 30BF   | PRI=0                                                     | 269     |
|      |             | STR D,12,0+PRI                                            | 270     |
|      |             | ITERATE 4 TIMES                                           | 271     |
|      |             | PRI=@PRI+@PI(.@K.)                                        | 272     |
|      |             | LDR IS,5,-5                                               | 273     |
|      |             | BRC I,7,a16                                               | 274     |
| 0067 | 4FB5        | EQU \$                                                    | 275     |
| 0068 | C107 0077   | D,6,0+K                                                   | 276     |
|      |             | LDR D,7,0+PRI                                             | 277     |
| 006A | E206 31B4   | ADD DX,7,0+PRI,6                                          | 278     |
| 006C | E207 30BF   | STR D,7,0+PRI                                             | 279     |
| 006E | 8367 30B9   | END                                                       | 280     |
| 0070 | 9C07 30EF   | EQU \$                                                    | 281     |
|      |             | IBN I,5,a15                                               | 282     |
| 0072 | E207 31B4   | LDR D,7,0+K                                               | 283     |
| 0074 | 6017        | ADD IS,7,1                                                | 284     |
| 0075 | 9C07 31B4   | STR D,7,0+K                                               | 285     |
| 0077 | C905 006A   | END                                                       | 286     |



| LUC  | OBJECT CODE | CARD IMAGE                        | CARDNUM |
|------|-------------|-----------------------------------|---------|
| 0079 | E207 30BF   | PRITEM=@PRI                       | 267     |
| 0078 | 9C07 31B8   | LDR D,7,0+PRI                     | 268     |
|      |             | STR D,7,0+PRITEM                  | 269     |
|      |             | FFI=@PRI/(4+@MP)                  | 270     |
| 007D | E207 31B5   | LDR D,7,0+MP                      | 271     |
| 007F | 6047 30BF   | ADD IS,7,4                        | 272     |
| 0080 | E203 30BF   | LDR D,3,0+PRI                     | 273     |
| 0082 | D8F2        | SHD RA,2,16                       | 274     |
| 0083 | FC72        | DIV R,2,7                         | 275     |
| 0084 | 9C03 30BF   | STR D,3,0+PRI                     | 276     |
|      |             | TEST @PRITEM MOD(4+@MP)>(4+@MP)/2 | 277     |
| 0086 | E202 31B5   | LDR D,2,0+MP                      | 278     |
| 0088 | 6042        | ADD IS,2,4                        | 279     |
| 0089 | E204 31B8   | LDR D,4,0+PRITEM                  | 280     |
| 008B | D8F3        | SHD RA,3,16                       | 281     |
| 008C | FC23        | DIV R,3,2                         | 282     |
| 008D | E204 31B5   | LDR D,4,0+MP                      | 283     |
| 008F | 6044        | ADD IS,4,4                        | 284     |
| 0090 | E045        | LDR R,5,4                         | 285     |
| 0091 | D8F4        | SHD RA,4,16                       | 286     |
| 0092 | 4026        | LDR IS,6,2                        | 287     |
| 0093 | FC04        | DIV R,4,6                         | 288     |
| 0094 | B053        | CMP R,3,5                         | 289     |
|      |             | IFSO: PRI=@PRI+1                  | 290     |
| 0095 | C106 009C   | EQU 1,6,@20                       | 291     |
| 0097 | E206 30BF   | BRC D,6,0+PRI                     | 292     |
| 0099 | 6016        | LDR IS,6,1                        | 293     |
| 009A | 9C06 30BF   | ADD STR D,6,0+PRI                 | 294     |
|      |             | END EQU \$                        | 295     |
| 009C | E367 2FA0   | EQU @20                           | 296     |
| 009D | 9C07 31B3   | EQU @17                           | 297     |
|      |             | DP=@NS(.@I.)                      | 298     |
|      |             | LDR R,6,10                        | 299     |
|      |             | LDR DX,7,0+NS,6                   | 300     |
|      |             | STR D,7,0+DP                      | 301     |
|      |             | CNDX=@CNDX+6                      | 302     |
|      |             | LDR D,7,0+CNDX                    | 303     |
|      |             | ADD IS,7,6                        | 304     |
|      |             | STR D,7,0+CNDX                    | 305     |
|      |             | CINDX(.@I/5.)=@CNDX               | 306     |
|      |             | LDR R,8,10                        | 307     |
|      |             | SHD RA,7,16                       | 308     |
|      |             | LDR IS,2,5                        | 309     |
|      |             | DIV R,7,2                         | 310     |
|      |             | LDR D,2,0+CNDX                    | 311     |
|      |             | STR DX,2,0+CINDX,8                | 312     |
|      |             | CHAN(.(@CNDX+1).)=@BUF(.(@DP+1).) | 313     |
|      |             | LDR D,2,0+CNDX                    | 314     |
|      |             | ADD IS,2,1                        | 315     |
| 00A6 | E0A8        |                                   | 316     |
| 00A7 | D8F7        |                                   | 317     |
| 00A8 | 4052        |                                   | 318     |
| 00A9 | FC27        |                                   | 319     |
| 00AA | E202 30C1   |                                   | 320     |
| 00AC | 9B82 31B8   |                                   | 321     |
| 00AE | E202 30C1   |                                   | 322     |
| 00B0 | 6012        |                                   | 323     |



| LOC  | OBJECT CODE | CARD IMAGE                                                 | CARDNUM |
|------|-------------|------------------------------------------------------------|---------|
| 00B1 | E203 31B3   | LDR D,3,0+DP                                               | 320     |
| 00B3 | E013        | IS,3,1                                                     | 321     |
| 00B4 | E334 2000   | DX,4,0+BUF,3                                               | 322     |
| 00B6 | 9B24 30C2   | LDR DX,4,0+CHAN,2                                          | 323     |
|      |             | CHAN(,@CNDX+2,)=@IMIN(,@I/5*2,.)                           | 324     |
| 00B8 | E204 30C1   | LDR D,4,0+CNDX                                             | 325     |
| 00BA | 6024        | IS,4,2                                                     | 326     |
| 00BB | E0A6        | R,6,10                                                     | 327     |
| 00BC | D8F5        | RA,5,16                                                    | 328     |
| 00BD | 4057        | IS,7,5                                                     | 329     |
| 00BE | FC75        | R,5,7                                                      | 330     |
| 00BF | CD06        | I,6,2                                                      | 331     |
| 00C1 | E367 30B8   | DX,7,0+TMIN,6                                              | 332     |
| 00C3 | 9B47 30C2   | STR DX,7,0+CHAN,4                                          | 333     |
|      |             | CHAN(,@CNDX+3,)=@IMIN(,@I/5*2+1,.)                         | 334     |
| 00C5 | E207 30C1   | LDR D,7,0+CNDX                                             | 335     |
| 00C7 | 6037        | IS,7,3                                                     | 336     |
| 00C8 | E0A3        | R,3,10                                                     | 337     |
| 00C9 | D8F2        | RA,2,16                                                    | 338     |
| 00CA | 4054        | IS,4,5                                                     | 339     |
| 00CB | FC42        | R,2,4                                                      | 340     |
| 00CC | CD03        | I,3,2                                                      | 341     |
| 00CE | 6013        | IS,3,1                                                     | 342     |
| 00CF | E334 30B8   | DX,4,0+TMIN,3                                              | 343     |
| 00D1 | 9B74 30C2   | STR DX,4,0+CHAN,7                                          | 344     |
|      |             | CHAN(,@CNDX+4,)=@BUF(,@DP+4,.)                             | 345     |
| 00L3 | E204 30C1   | LDR D,4,0+CNDX                                             | 346     |
| 00L5 | 6044        | IS,4,4                                                     | 347     |
| 00L6 | E205 31B3   | ADD D,5,0+DP                                               | 348     |
| 00L8 | 6045        | IS,5,4                                                     | 349     |
| 00D9 | E356 2000   | DX,6,0+BUF,5                                               | 350     |
| 00LB | 9B46 30C2   | LDR DX,6,0+CHAN,4                                          | 351     |
|      |             | CHAN(,@CNDX+5,)=@PRI                                       | 352     |
| 00DD | E206 30C1   | STR D,6,0+CNDX                                             | 353     |
| 00DF | 6056        | LDR D,6,5                                                  | 354     |
| 00E0 | E207 30BF   | ADD IS,6,5                                                 | 355     |
| 00E2 | 9B67 30C2   | LDR D,7,0+PRI                                              | 356     |
|      |             | STR DX,7,0+CHAN,6                                          | 357     |
|      |             | FLAG PULSE TRAIN AS STABLE AND INCREMENT CHANNEL COUNTER " | 358     |
|      |             | SSF(,@I/5,)=TRUE                                           | 359     |
| 00E4 | E0A8        | LDR R,8,10                                                 | 360     |
| 00E5 | D8F7        | RA,7,16                                                    | 361     |
| 00E6 | 4052        | IS,2,5                                                     | 362     |
| 00E7 | FC27        | R,2,2                                                      | 363     |
| 00E8 | 4012        | IS,2,1                                                     | 364     |
| 00E9 | 9B82 3163   | STR DX,2,0+SSF,8                                           | 365     |
|      |             | NC=@NC+1                                                   | 366     |
| 00EB | E202 30B8   | LDR D,2,0+NC                                               | 367     |
| 00ED | 6012        | IS,2,1                                                     | 368     |
| 00EE | 9C02 30B8   | ADD D,2,0+NC                                               | 369     |
|      |             | STR D,2,0+NC                                               | 370     |
| 00F0 | BF17        | RETURN R,7,1                                               | 371     |
|      |             | BFC                                                        | 372     |



TIME: 22:58:25

03/18/75

PAGE

CARDNUM

```

LOC      OBJECT CODE      CARD IMAGE
00F1      EQA3
00F2      D8F2
00F3      4054
00F4      FC42
00F5      4014
00F6      9834 313B
00F8      BF17

      .END LOOP1
      EQU $
      ./" TRAIN FAILED STABILITY TEST. FLAG TRAIN AS NON STABLE "/
      .
      . NSSF(.@I/5).)=TRUE
      .   LDR R,3,10
      .   SHD RA,2,16
      .   LDR IS,4,5
      .   DIV R,2,4
      .   LDR IS,4,1
      .   STK DX,4,0+NSSF,3
      .END PRICOMP
      BRC R,7,1

      .
      .PROCEDURE MAIN
      .ENTRY MAIN
      ./" DESCRIPTION OF VARIABLES
      .
      . BRNGF RCVR ANTENNA BEARING AT TOA OF FIRST PULSE OF A SIGNAL
      . BRN3L RCVR ANTENNA BEARING AT TOA OF LAST PULSE OF A TRAIN
      . CON1 DOUBLE PRECISION TIME CALCULATION VARIABLE
      . CON2 DOUBLES PRECISION TIME CALCULATION VARIABLE
      . II,JJ INDICES
      . SCANTM RCVR ANTENNA SCAN TIME FOR ONE BEAMWIDTH
      . TOAL LOWER HALF DOUBLE PRECISION TIME OF ARRIVAL
      . TOAU UPPER HALF DOUBLE PRECISION TIME OF ARRIVAL
      . TMAX EXPECTED TIME OF LAST PULSE INTERCEPT
      . TMIN DOUBLE PRECISION TIME OF ARRIVAL OF FIRST PULSE
      . ITEM DOUBLE PRECISION TIME CALCULATION VARIABLE
      . TTEST DOUBLE PRECISION TIME CALCULATION VARIABLE
      ."/
      .DECLARE(BRNGF,BRNGL(39),SCANTM(1),II,JJ,STR)RAM
      BRNGF ORG RAM
      BRNGL DS 1 40
      SCANTM DS 2 2
      II DS 1 1
      JJ DS 1 1
      STR DS 1 1
      .DECLARE(TOAL,TOAU,TMAX(79),TTEST(1),ITEM(1))RAM
      TOAL DS 1 1
      TOAU DS 1 80
      TMAX DS 2 2
      TTEST DS 2 2
      .DECLARE(CON1(1),CON2(1))RAM
      CON1 DS 2 2
      CON2 DS 2 2

```

373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425



| LOC | OBJECT CODE | CARD IMAGE                                            | CARDNUM |
|-----|-------------|-------------------------------------------------------|---------|
|     |             | ./" INITIALIZE VARIABLES "/                           |         |
|     |             | : CNDX=-6                                             | 426     |
|     |             | ORG RQM                                               | 427     |
|     |             | EQU \$                                                | 428     |
|     |             | LDR IS,12,0                                           | 429     |
|     |             | LDR IS,2,-6                                           | 430     |
|     |             | STR D,2,0+CNDX                                        | 431     |
|     |             | SCANTM=1120                                           | 432     |
|     |             | LDR I,2,1120                                          | 433     |
|     |             | STR D,2,0+SCANTM                                      | 434     |
|     |             | I=0                                                   | 435     |
|     |             | LDR R,10,12                                           | 436     |
|     |             | BP=0                                                  | 437     |
|     |             | LDR R,11,12                                           | 438     |
|     |             | NC=1                                                  | 439     |
|     |             | LDR IS,2,1                                            | 440     |
|     |             | STR D,2,0+NC                                          | 441     |
|     |             | ./" SET NEW SIGNAL POINTER FILE TO NEGATIVE VALUES "/ | 442     |
|     |             | : ITERATE 200 TIMES                                   | 443     |
|     |             | NS(.AI.)=-1                                           | 444     |
|     |             | LDR I,2,-201                                          | 445     |
|     |             | BRC I,7,@22                                           | 446     |
|     |             | EQU \$                                                | 447     |
|     |             | LDR R,3,10                                            | 448     |
|     |             | LDR IS,4,-1                                           | 449     |
|     |             | STR DX,4,0+NS,3                                       | 450     |
|     |             | I=@1+1                                                | 451     |
|     |             | LDR R,4,10                                            | 452     |
|     |             | ADD IS,4,1                                            | 453     |
|     |             | LDR R,10,4                                            | 454     |
|     |             | END                                                   | 455     |
|     |             | @22 EQU \$                                            | 456     |
|     |             | IBN I,2,@21                                           | 457     |
|     |             | I=0                                                   | 458     |
|     |             | LDR R,10,12                                           | 459     |
|     |             | NS(.AI.)=0                                            | 460     |
|     |             | LDR R,4,10                                            | 461     |
|     |             | STR DX,12,0+NS,4                                      | 462     |
|     |             | ./" LOAD INPUT BUFFER FROM CARDS "/                   | 463     |
|     |             | : READ_DEC(BUF,BUF+3999)                              | 464     |
|     |             | SIM I,@23                                             | 465     |
|     |             | BRC IS,7,@24                                          | 466     |
|     |             | EQU \$                                                | 467     |
|     |             | DC O+BUF                                              | 468     |
|     |             | DC 3999+BUF                                           | 469     |
|     |             | EQU \$                                                | 470     |
|     |             | @23                                                   | 471     |
|     |             | @24                                                   | 472     |
|     |             | FE01 011A                                             | 473     |
|     |             | 0017                                                  | 474     |
|     |             | 011A 2000                                             | 475     |
|     |             | 011B 2F5F                                             | 476     |
|     |             |                                                       | 477     |
|     |             |                                                       | 478     |



| LOC  | OBJECT CODE | CARD IMAGE                                                      | CARDNUM |
|------|-------------|-----------------------------------------------------------------|---------|
| 011C | E0B4        | MAINLP:                                                         | 479     |
| 011D | 6024        | DO EQU \$                                                       | 480     |
| 011E | E345        | 225                                                             | 481     |
| 0120 | 9C05        | 20C0 31E7                                                       | 482     |
| 0122 | E0B5        | LOAD OP TOA OF FIRST PULSE OF SIGNAL FROM INPUT BUFFER AND      | 483     |
| 0123 | 6035        | COMPUTE EXPECTED TIME OF ARRIVAL OF LAST PULSE BASED ON RCVR    | 484     |
| 0124 | E356        | ANTENNA SCAN TIME FOR ONE ANTENNA BEAMWIDTH "/                  | 485     |
| 0126 | 9C06        | 2000 31E8                                                       | 486     |
| 0128 | BC1E        | TOAL=2BUF(.12BP+2).)                                            | 487     |
| 012A | A90E        | LDR R,4,11                                                      | 488     |
| 012C | 9C1E        | ADD IS,4,2                                                      | 489     |
|      |             | LDR DX,5,0+BUF,4                                                | 490     |
|      |             | STR D,5,0+TOAL                                                  | 491     |
|      |             | TOAU=2BUF(.12BP+3).)                                            | 492     |
|      |             | LDR R,5,11                                                      | 493     |
|      |             | ADD IS,5,3                                                      | 494     |
|      |             | LDR DX,6,0+BUF,5                                                | 495     |
|      |             | STR D,6,0+TOAU                                                  | 496     |
|      |             | CODE                                                            | 497     |
|      |             | LORM D,14,TOAL,2                                                | 498     |
|      |             | DADD D,14,SCANTM                                                | 499     |
|      |             | STRM D,14,ITEM,2                                                | 500     |
|      |             | END                                                             | 501     |
|      |             | STR=2I/5*2                                                      | 502     |
| 012E | E0A7        | R,7,10                                                          | 503     |
| 012F | D8F6        | RA,6,16                                                         | 504     |
| 0130 | 4058        | IS,8,5                                                          | 505     |
| 0131 | FC86        | IS,6,8                                                          | 506     |
| 0132 | CD07        | I,7,2                                                           | 507     |
| 0134 | 5C07        | MUL I,7,2                                                       | 508     |
|      |             | STR D,7,0+STR                                                   | 509     |
|      |             | TMAX(.12STR)=2ITEM                                              | 510     |
|      |             | LDR D,8,0+STR                                                   | 511     |
|      |             | LDR D,2,0+ITEM                                                  | 512     |
|      |             | STR DX,2,0+TMAX,8                                               | 513     |
|      |             | TMAX(.12STR+1)=2ITEM(.1)                                        | 514     |
|      |             | LDR D,2,0+STR                                                   | 515     |
|      |             | ADD IS,2,1                                                      | 516     |
|      |             | LDR D,3,1+ITEM                                                  | 517     |
|      |             | STR DX,3,0+TMAX,2                                               | 518     |
|      |             | TMIN(.12STR)=2TOAL                                              | 519     |
|      |             | LDR D,3,0+STR                                                   | 520     |
|      |             | LDR D,4,0+TOAL                                                  | 521     |
|      |             | STR DX,4,0+TMIN,3                                               | 522     |
|      |             | TMIN(.12STR+1)=2TOAU                                            | 523     |
|      |             | LDR D,4,0+STR                                                   | 524     |
|      |             | ADD IS,4,1                                                      | 525     |
|      |             | LDR D,5,0+TOAU                                                  | 526     |
|      |             | STR DX,5,0+TMIN,4                                               | 527     |
|      |             | COMPUTE:                                                        | 528     |
|      |             | DO EQU \$                                                       | 529     |
|      |             | 227                                                             | 530     |
|      |             | "/ INCREMENT BUFFER POINTER AND CHECK AT BUFFER LIMIT. IFSC, GO | 531     |



| LCC  | OBJECT CODE | CARD IMAGE                                                         | CARDNUM |
|------|-------------|--------------------------------------------------------------------|---------|
|      |             | COMPRESS RESIDUAL PULSES WITHIN INPUT BUFFER FOR FUTHER PROCESSING |         |
| 0150 | E0B5        | BP=BBP+5                                                           | 532     |
| 0151 | E055        | LDR R,5,11                                                         | 533     |
| 0152 | E058        | ADD R,5,5                                                          | 534     |
| 0153 | E0B5        | LDR R,11,5                                                         | 535     |
| 0154 | B105 0F9F   | TEST BBP>3999                                                      | 536     |
|      |             | LDR R,5,11                                                         | 537     |
|      |             | CMP I,5,3999                                                       | 538     |
| 0150 | C106 015A   | IFSO: LEAVE MAINLP                                                 | 539     |
| 0158 | C107 02CD   | EQU \$                                                             | 540     |
|      |             | BRC I,6,032                                                        | 541     |
|      |             | END I,7,026                                                        | 542     |
|      |             | EQU \$                                                             | 543     |
|      |             | EQU \$                                                             | 544     |
|      |             | IF NO SIGNAL IN BUFFER INDEXED BY BUFFER POINTER INCREMENT         | 545     |
|      |             | POINTER AND REPEAT COMPUTE "/                                      | 546     |
|      |             | TEST BBP(.BBP.)                                                    | 547     |
| 015A | E0B5        | LDR R,5,11                                                         | 548     |
| 015B | E356 2000   | LDR DX,6,0+BUF,5                                                   | 549     |
|      |             | ZERO: REPEAT COMPUTE                                               | 550     |
| 015D | C105 0161   | EQU \$                                                             | 551     |
| 015F | C107 0150   | BRC I,5,036                                                        | 552     |
|      |             | END I,7,027                                                        | 553     |
|      |             | EQU \$                                                             | 554     |
|      |             | EQU \$                                                             | 555     |
|      |             | I=0                                                                | 556     |
| 0161 | E0CA        | LDR R,10,12                                                        | 557     |
| 0162 | E0CS        | LDR R,9,12                                                         | 558     |
|      |             | SORT:                                                              | 559     |
|      |             | DO                                                                 | 560     |
|      |             | IF AT END OF NEW SIGNAL POINTER FILE, LEAVE SORT "/                | 561     |
|      |             | TEST 01>195                                                        | 562     |
| 0163 | E0A6        | LDR R,6,10                                                         | 563     |
| 0164 | B106 00C3   | CMP I,6,195                                                        | 564     |
|      |             | IFSO: LEAVE SORT                                                   | 565     |
|      |             | EQU \$                                                             | 566     |
| 0166 | C106 016A   | BRC I,6,042                                                        | 567     |
| 0168 | C107 02B3   | END I,7,038                                                        | 568     |
|      |             | EQU \$                                                             | 569     |
|      |             | EQU \$                                                             | 570     |
|      |             | IF NEW SIGNAL POINTER FILE INDEXED BY I IS EMPTY INDEX AND         | 571     |
|      |             | "/                                                                 | 572     |
|      |             | END                                                                | 573     |
|      |             | EQU \$                                                             | 574     |
|      |             | EQU \$                                                             | 575     |
|      |             | IF NEW SIGNAL POINTER FILE INDEXED BY I IS EMPTY INDEX AND         | 576     |
|      |             | "/                                                                 | 577     |
|      |             | END                                                                | 578     |
|      |             | EQU \$                                                             | 579     |
|      |             | EQU \$                                                             | 580     |
|      |             | IF NEW SIGNAL POINTER FILE INDEXED BY I IS EMPTY INDEX AND         | 581     |
|      |             | "/                                                                 | 582     |
|      |             | END                                                                | 583     |
|      |             | EQU \$                                                             | 584     |
|      |             | EQU \$                                                             | 585     |



| LOC  | OBJECT CODE | CARD IMAGE                                                    | CARDNUM |
|------|-------------|---------------------------------------------------------------|---------|
| 016A | E0A6        | TEST @NS(.@I.)                                                | 585     |
| 016B | E367 2FA0   | LDR R,6,10                                                    | 586     |
| 016D | C103 0174   | LDR DX,1,0+NS,6                                               | 587     |
| 016F | E0A7        | NEG: DO                                                       | 588     |
| 0170 | E057        | EQU \$                                                        | 589     |
| 0171 | E07A        | BRC 1,3,@40                                                   | 590     |
| 0172 | C107 0163   | I=@I+5                                                        | 591     |
|      |             | LDR R,7,10                                                    | 592     |
|      |             | ADD IS,7,5                                                    | 593     |
|      |             | LDR R,10,7                                                    | 594     |
|      |             | REPEAT SORT                                                   | 595     |
|      |             | BRC 1,7,@37                                                   | 596     |
|      |             | END                                                           | 597     |
|      |             | END                                                           | 598     |
|      |             | EQU \$                                                        | 599     |
|      |             | EQU \$                                                        | 600     |
|      |             | /"                                                            | 601     |
|      |             | COMPARE TOA OF BUFFER SIGNAL WITH TMAX OF SIGNAL FOUND IN NEW | 602     |
|      |             | SIGNAL POINTER FILE. IF TOA OF BUFFER SIGNAL IS GREATER THAN  | 603     |
|      |             | CR EQUAL TO TMAX, TEST FOR STABLE SIGNAL IN NS POINTER FILE.  | 604     |
|      |             | IF SIGNAL HAS BEEN CLASSIFIED STABLE, RESET ACTIVE CHANNEL    | 605     |
|      |             | COUNTER AND SSF FLAG, ZERO FIRST FIVE PULSES FROM THIS SIGNAL | 606     |
|      |             | IN INPUT BUFFER AND COMPUTE EMITTER BEARING USING FIRST AND   | 607     |
|      |             | DOA OF SIGNAL "/                                              | 608     |
|      |             | CON1=@BUF(.(@BP+2).)                                          | 609     |
| 0174 | E0B7        | LDR R,7,11                                                    | 610     |
| 0175 | 6C27        | ADD IS,7,2                                                    | 611     |
| 0176 | E378 2000   | LDR DX,8,0+BUF,7                                              | 612     |
| 0178 | 9C08 323D   | STR D,8,0+CON1                                                | 613     |
|      |             | CCN1(.1.)=@BUF(.(@BP+3).)                                     | 614     |
| 017A | E0B8        | LDR R,8,11                                                    | 615     |
| 017B | 6C38        | ADD IS,8,3                                                    | 616     |
| 017C | E382 2000   | LDR CX,2,0+BUF,8                                              | 617     |
| 017E | 9C02 323E   | STR D,2,1+CON1                                                | 618     |
|      |             | CON2=@TMAX(.(@I/5*2).)                                        | 619     |
| 018J | E0A3        | LDR R,3,10                                                    | 620     |
| 018J | 08F2        | SHD RA,2,16                                                   | 621     |
| 0182 | 4054        | LDR IS,4,5                                                    | 622     |
| 0183 | FC42        | DIV R,2,4                                                     | 623     |
| 0184 | C003        | MUL I,3,2                                                     | 624     |
| 0186 | E334        | LDR DX,4,0+TMAX,3                                             | 625     |
| 0188 | 9C04 323F   | STR D,4,0+CON2                                                | 626     |
|      |             | CON2(.1.)=@TMAX(.(@I/5*2+1).)                                 | 627     |
| 018A | E0A5        | LDR R,5,10                                                    | 628     |
| 018B | 08F4        | SHD RA,4,16                                                   | 629     |
| 018C | 4056        | LDR IS,6,5                                                    | 630     |
| 018D | FC54        | DIV R,4,6                                                     | 631     |
| 018E | C005        | MUL I,5,2                                                     | 632     |
| 0190 | 6015        | LDR DX,6,0+TMAX,5                                             | 633     |
| 0191 | E356 31E9   | ADD IS,5,1                                                    | 634     |
|      |             | LDR DX,6,0+TMAX,5                                             | 635     |



| LOC  | OBJECT CODE | CARD IMAGE                     | CARDNUM |
|------|-------------|--------------------------------|---------|
| 0153 | 9C06 3240   | STR D,6,1+CON2                 | 638     |
| 0195 | BC1E 323D   | LDRM D,14,CON1,2               | 639     |
| 0197 | BC17 323F   | LDRM D,7,CON2,2                | 640     |
| 0199 | AAE7        | DSUB R,7,14                    | 641     |
| 019A | 9C17 3239   | STRM D,7,1TEST,2               | 642     |
| 019C | E206 3239   | END TEST                       | 643     |
| 019E | E206 3239   | TEST                           | 644     |
| 019F | E206 3239   | TEST                           | 645     |
| 01A0 | E0A7        | NEG,ZERO: DO                   | 646     |
| 01A1 | D8F6        | EQU \$                         | 647     |
| 01A2 | 4058        | BRC I,1,250                    | 648     |
| 01A3 | FC86        | TEST @SSF(,(@I/5).)            | 649     |
| 01A4 | E378 3163   | LDR R,7,10                     | 650     |
| 01A6 | C106 01EE   | SHD RA,6,16                    | 651     |
| 01A8 | E208 30B8   | LDR IS,8,5                     | 652     |
| 01AA | GFF8        | DIV R,6,8                      | 653     |
| 01AB | 9C08 30B8   | LDR DX,8,0+SSF,7               | 654     |
| 01AD | E0A3        | PGS: DO                        | 655     |
| 01AE | D8F2        | EQU \$                         | 656     |
| 01AF | 4054        | BRC I,6,254                    | 657     |
| 01B0 | FC42        | LDK D,8,0+NC                   | 658     |
| 01B1 | 983C 3163   | ADD IS,8,-1                    | 659     |
| 01B3 | E0A4        | STR C,8,0+NC                   | 660     |
| 01B4 | E345        | SSF(,(@I/5).)=FALSE            | 661     |
| 01B5 | E356 2000   | LDR R,3,10                     | 662     |
| 01B8 | 9C06 31B9   | SHD RA,2,16                    | 663     |
| 01BA | 9C0C 31E4   | LDR IS,4,5                     | 664     |
| 01BC | 4FA6        | DIV R,2,4                      | 665     |
| 01BD | C107 01D9   | STR DX,12,0+SSF,3              | 666     |
| 01BF | 9C0C 31E5   | LDR R,4,10                     | 667     |
| 01C1 | 4FA7        | LDR DX,5,0+NS,4                | 668     |
| 01C2 | C107 01D2   | LDR DX,6,0+BUF,5               | 669     |
| 01C4 | E0A8        | STR D,6,0+BRNGF                | 670     |
| 01C5 | 82C8 31E4   | I=0                            | 671     |
| 01C7 | E362 2FA0   | ITERATE 5 TIMES                | 672     |
| 01C9 | 8202 31E5   | BUF(,(@NS(,(@I+@II).)+@JJ).)=0 | 673     |
|      |             | ITERATE 5 TIMES                | 674     |
|      |             | BUF(,(@NS(,(@I+@II).)+@JJ).)=0 | 675     |
|      |             | ITERATE 5 TIMES                | 676     |
|      |             | BUF(,(@NS(,(@I+@II).)+@JJ).)=0 | 677     |
|      |             | ITERATE 5 TIMES                | 678     |
|      |             | BUF(,(@NS(,(@I+@II).)+@JJ).)=0 | 679     |
|      |             | ITERATE 5 TIMES                | 680     |
|      |             | BUF(,(@NS(,(@I+@II).)+@JJ).)=0 | 681     |
|      |             | ITERATE 5 TIMES                | 682     |
|      |             | BUF(,(@NS(,(@I+@II).)+@JJ).)=0 | 683     |
|      |             | ITERATE 5 TIMES                | 684     |
|      |             | BUF(,(@NS(,(@I+@II).)+@JJ).)=0 | 685     |
|      |             | ITERATE 5 TIMES                | 686     |
|      |             | BUF(,(@NS(,(@I+@II).)+@JJ).)=0 | 687     |
|      |             | ITERATE 5 TIMES                | 688     |
|      |             | BUF(,(@NS(,(@I+@II).)+@JJ).)=0 | 689     |
|      |             | ITERATE 5 TIMES                | 690     |
|      |             | BUF(,(@NS(,(@I+@II).)+@JJ).)=0 | 691     |







CARDNUM

CARD IMAGE

LCC OBJECT CODE

```

01F7 C107 0204      a59      BRC      I,7,@60
01F9 E0A5           EQU      $
01FA 8205 31E4      LDR      R,5,10
01FC 4FF6           ADD      D,5,0+II
01FD 9856 2FA0      LDR      IS,6,-1
                        STR      DX,6,0+NS,5
                        II=@II+1
01FF E206 31E4      LDR      0,6,0+II
0201 6016           ADD      IS,6,1
0202 9C06 31E4      STR      D,6,0+II
                        END
0204 C904 01F9      EQU      $
                        IBN      I,4,@59
                        I=@I+5
0206 E0A6           LDR      R,6,10
0207 6056           ADD      IS,6,5
0208 E06A           LDR      R,10,6
                        REPEAT SORT
0209 C107 0163      BRC      I,7,@37
                        END
020B E0B6           EQU      $
020C 6016           EQU      @BUF(.(@BP+1).)=@BUF(.(@NS(.(@I.)+1).))
020D E0A7           LDR      R,6,11
020E 6018           ADD      IS,6,1
0210 6018           LDR      DX,8,0+NS,7
0211 E362 2000      ADD      IS,8,1
0213 B382 2000      LDR      DX,2,0+BUF,6
                        CMP      DX,2,0+BUF,8
                        IFNOT: DO
0215 C102 021C      EQU      $
                        BRC      I,2,@64
                        I=@I+5
0217 E0A2           LDR      R,2,10
0218 6052           ADD      IS,2,5
0219 E02A           LDR      R,10,2
                        REPEAT SORT
021A C107 0163      BRC      I,7,@37
                        END
021C E0B2           EQU      $
021D 6061           EQU      $
                        END
021E 6061           EQU      $
                        TEST @BUF(.(@BP+4).)=@BUF(.(@NS(.(@I.)+4).))
                        LDR      R,2,11
021F 6061           EQU      $
0220 6061           EQU      $
0221 6061           EQU      $
0222 6061           EQU      $
0223 6061           EQU      $
0224 6061           EQU      $
0225 6061           EQU      $
0226 6061           EQU      $
0227 6061           EQU      $
0228 6061           EQU      $
0229 6061           EQU      $
0230 6061           EQU      $
0231 6061           EQU      $
0232 6061           EQU      $
0233 6061           EQU      $
0234 6061           EQU      $
0235 6061           EQU      $
0236 6061           EQU      $
0237 6061           EQU      $
0238 6061           EQU      $
0239 6061           EQU      $
0240 6061           EQU      $
0241 6061           EQU      $
0242 6061           EQU      $
0243 6061           EQU      $
0244 6061           EQU      $
0245 6061           EQU      $
0246 6061           EQU      $
0247 6061           EQU      $
0248 6061           EQU      $
0249 6061           EQU      $
0250 6061           EQU      $
0251 6061           EQU      $
0252 6061           EQU      $
0253 6061           EQU      $
0254 6061           EQU      $
0255 6061           EQU      $
0256 6061           EQU      $
0257 6061           EQU      $
0258 6061           EQU      $
0259 6061           EQU      $
0260 6061           EQU      $
0261 6061           EQU      $
0262 6061           EQU      $
0263 6061           EQU      $
0264 6061           EQU      $
0265 6061           EQU      $
0266 6061           EQU      $
0267 6061           EQU      $
0268 6061           EQU      $
0269 6061           EQU      $
0270 6061           EQU      $
0271 6061           EQU      $
0272 6061           EQU      $
0273 6061           EQU      $
0274 6061           EQU      $
0275 6061           EQU      $
0276 6061           EQU      $
0277 6061           EQU      $
0278 6061           EQU      $
0279 6061           EQU      $
0280 6061           EQU      $
0281 6061           EQU      $
0282 6061           EQU      $
0283 6061           EQU      $
0284 6061           EQU      $
0285 6061           EQU      $
0286 6061           EQU      $
0287 6061           EQU      $
0288 6061           EQU      $
0289 6061           EQU      $
0290 6061           EQU      $
0291 6061           EQU      $
0292 6061           EQU      $
0293 6061           EQU      $
0294 6061           EQU      $
0295 6061           EQU      $
0296 6061           EQU      $

```

TEST FOR MATCH BETWEEN BUFFER SIGNAL FREQ AND PW AND NS SIGNAL  
 POINTER FILE FREQ AND PW. IF BOTH DO NOT MATCH REPEAT SORT.  
 IF MATCH OCCURS AND SIGNAL HAS BEEN CLASSIFIED AS STABLE, N  
 ZERO BUFFER SIGNAL AND REPEAT COMPUTE. IF SIGNAL HAS BEEN  
 CLASSIFIED AS NON STABLE, REPEAT COMPUTE. "/



| LOC  | OBJECT CODE | CARD IMAGE                | CARDNUM |
|------|-------------|---------------------------|---------|
| 021D | 6042        | ADD IS,2,4                | 797     |
| 021E | E0A3        | LDR R,3,10                | 798     |
| 021F | E334        | LDR DX,4,0+NS,3           | 799     |
| 0221 | 6044        | ADD IS,4,4                | 800     |
| 0222 | E325        | LDR DX,5,0+BUF,2          | 801     |
| 0224 | B343        | CMR DX,5,0+BUF,4          | 802     |
|      |             | IF NOT: DQ                | 803     |
| 0226 | C102 022D   | EQU \$                    | 804     |
|      |             | BRC I,2,@68               | 805     |
|      |             | I=@1+5                    | 806     |
| 0228 | E0A5        | LDR R,5,10                | 807     |
| 0229 | 6055        | ADD IS,5,5                | 808     |
| 022A | E05A        | LDR R,10,5                | 809     |
|      |             | REPEAT SORT               | 810     |
| 022B | C107 0163   | BRC I,7,@37               | 811     |
|      |             | END                       | 812     |
|      |             | END                       | 813     |
|      |             | EQU \$                    | 814     |
|      |             | EQU \$                    | 815     |
|      |             | TEST @SSF(,@1/5).)        | 816     |
| 022D | E046        | LDR R,6,10                | 817     |
| 022E | D8F5        | SHD RA,5,16               | 818     |
| 022F | 4057        | LDR IS,7,5                | 819     |
| 0230 | FC75        | DIV R,5,7                 | 820     |
| 0231 | E367        | LDR DX,7,0+SSF,6          | 821     |
|      |             | NCNZERO: DU               | 822     |
| 0233 | C102 C251   | EQU \$                    | 823     |
|      |             | BRC I,2,@72               | 824     |
|      |             | BRNGL(,@1/5.)=@BUF(,@BP.) | 825     |
| 0235 | E0A8        | LDR R,8,10                | 826     |
| 0236 | D8F7        | SHD RA,7,16               | 827     |
| 0237 | 4052        | LDR IS,2,5                | 828     |
| 0238 | FC27        | DIV R,7,2                 | 829     |
| 0239 | E0B2        | LDR R,2,11                | 830     |
| 023A | E323        | LDR DX,3,0+BUF,2          | 831     |
| 023C | 9B33        | STR DX,3,0+BRNGL,8        | 832     |
|      |             | I1=0                      | 833     |
| 023E | 9C0C 31E4   | STR D,12,0+I1             | 834     |
|      |             | ITERATE 5 TIMES           | 835     |
|      |             | BUF(,@BP+@I1.)=0          | 836     |
| 0240 | 4FA3        | LDR IS,3,-6               | 837     |
| 0241 | C107 0240   | EQU \$                    | 838     |
|      |             | I1,@74                    | 839     |
| 0243 | E0B4        | LDR R,4,11                | 840     |
| 0244 | E204        | ADD D,4,0+I1              | 841     |
| 0246 | 9B4C 31E4   | STR DX,12,0+BUF,4         | 842     |
|      |             | I1=@I1+1                  | 843     |
| 0248 | E204 31E4   | LDR D,4,0+I1              | 844     |
| 024A | 6014        | ADD IS,4,1                | 845     |
| 024B | 9C04 31E4   | STR D,4,0+I1              | 846     |
|      |             | END                       | 847     |
|      |             | EQU \$                    | 848     |
| 024D | C903 0243   | IBN I,3,@73               | 849     |







| LUC  | OBJECT CODE | CARD | IMAGE                                                              | CARDNUM |
|------|-------------|------|--------------------------------------------------------------------|---------|
| 026E | C906 025F   | .    | NS(. (a1+4) .) = @BP                                               | 903     |
| 0270 | E0A8        | .    | LDR R, 6, a79                                                      | 904     |
| 0271 | 6048        | .    | ADD R, 8, 10                                                       | 905     |
| 0272 | 9B8B 2FA0   | .    | IS, 8, 14                                                          | 906     |
| 0274 | E208 30B8   | .    | STR DX, 11, 0+NS, 8                                                | 907     |
| 0276 | 20A8        | .    | TEST @NC = 10                                                      | 908     |
|      |             | .    | LDR D, 8, 0+NC                                                     | 909     |
|      |             | .    | CMP IS, 8, 10                                                      | 910     |
|      |             | .    | IFSU: DO                                                           | 911     |
| 0277 | C105 0282   | .    | EQU \$                                                             | 912     |
|      |             | .    | BRC 1, 5, a88                                                      | 913     |
| 0279 | E0A3        | .    | NSSF(. (a1/5) .) = TRUE                                            | 914     |
| 027A | 0BF2        | .    | LDR R, 3, 10                                                       | 915     |
| 027B | 4054        | .    | SHD RA, 2, 16                                                      | 916     |
| 027C | FC42        | .    | LDR IS, 4, 5                                                       | 917     |
| 027D | 4014        | .    | OIV R, 2, 4                                                        | 918     |
| 027E | 5B34 313B   | .    | LDR IS, 4, 1                                                       | 919     |
|      |             | .    | STR DX, 4, 0+NSSF, 3                                               | 920     |
| 0280 | C107 0150   | .    | REPEAT COMPUTE                                                     | 921     |
|      |             | .    | BRC 1, 7, a27                                                      | 922     |
|      |             | .    | END                                                                | 923     |
|      |             | .    | END                                                                | 924     |
|      |             | .    | EQU \$                                                             | 925     |
|      |             | .    | EQU \$                                                             | 926     |
| 0282 | 4019        | .    | J=1                                                                | 927     |
|      |             | .    | ITERATE 4 TIMES                                                    | 928     |
| 0283 | 4FB4        | .    | PI(.aJ.) = @BUF(. (a1+aJ) .) + 2 .) - @BUF(. (a1+aJ-1) .) + 2 .) . | 929     |
| 0284 | C107 02A8   | .    | LDR IS, 9, 1                                                       | 930     |
|      |             | .    | LDR IS, 4, -5                                                      | 931     |
|      |             | .    | BRC 1, 7, a90                                                      | 932     |
| 0286 | E095        | .    | EQU \$                                                             | 933     |
| 0287 | E0A6        | .    | LDR R, 5, 9                                                        | 934     |
| 0288 | 8096        | .    | LDR R, 6, 10                                                       | 935     |
| 0289 | E367 2FA0   | .    | ADD R, 6, 9                                                        | 936     |
| 028A | 6027        | .    | LDR DX, 7, 0+NS, 6                                                 | 937     |
| 028C | E0A8        | .    | ADD IS, 7, 2                                                       | 938     |
| 028D | 8058        | .    | LDR R, 8, 10                                                       | 939     |
| 028E | 0FF8        | .    | ADD R, 8, 9                                                        | 940     |
| 028F | E382        | .    | ADD IS, 8, -1                                                      | 941     |
| 0291 | 6022        | .    | LDR DX, 2, 0+NS, 8                                                 | 942     |
| 0292 | E373        | .    | ADD IS, 2, 2                                                       | 943     |
| 0293 | 8723        | .    | LDR DX, 3, 0+BUF, 7                                                | 944     |
| 0294 | 9B53 30B9   | .    | SUB DX, 3, 0+BUF, 2                                                | 945     |
|      |             | .    | STR @PI(.aJ.)                                                      | 946     |
| 0298 | E093        | .    | LDR R, 3, 9                                                        | 947     |
| 0299 | E335 50B9   | .    | LDR DX, 5, 0+PI, 3                                                 | 948     |
|      |             | .    | NEG: PI(.aJ.) = @PI(.aJ.) + 32767                                  | 949     |
|      |             | .    | EQU \$                                                             | 950     |
| 029B | C103 02A5   | .    | BRC 1, 3, a94                                                      | 951     |
| 029D | E095        | .    | LDR R, 5, 9                                                        | 952     |
| 029E | E096        | .    | LDR R, 0, 9                                                        | 953     |
| 029F | E367 30B9   | .    | LDR DX, 7, 0+PI, 6                                                 | 954     |



| LOC  | OBJECT CODE | CARD             | IMAGE               | CARDNUM |
|------|-------------|------------------|---------------------|---------|
| 02A1 | 8107 7FFF   |                  |                     | 956     |
| 02A3 | 9857 30B9   |                  | ADD I,7,32767       | 957     |
|      |             |                  | STR DX,7,0+PI,5     | 958     |
|      |             | END              | EQU \$              | 959     |
|      |             | END              | EQU \$              | 960     |
|      |             | J=J+1            |                     | 961     |
| 02A5 | E097        |                  | LDR R,7,9           | 962     |
| 02A6 | 6017        |                  | ADD IS,7,1          | 963     |
| 02A7 | E079        |                  | LDR R,9,7           | 964     |
|      |             | END              |                     | 965     |
| 02A8 | C904 0286   |                  | EQU \$ I,4,@89      | 966     |
|      |             | CALL             | PRICOMP()           | 967     |
| 02AA | 6010        |                  | ADD IS,0,1          | 968     |
| 02AB | 8DAC        |                  | STK DR,0,11         | 969     |
| 02AC | ED01 00G0   |                  | PR I,1,0+PRICOMP    | 970     |
| 02AE | 8EAO        |                  | RET DR,0,11         | 971     |
| 02AF | 6FF0        |                  | ADD IS,0,-1         | 972     |
| 02B0 | 5C07        |                  | LDR RX,7,0,12       | 973     |
|      |             | REPEAT           | COMPUTE I,7,@27     | 974     |
| 02B1 | C107 0150   |                  | BRC                 | 975     |
|      |             | END SORT         |                     | 976     |
|      |             | I=0              | EQU \$              | 977     |
| 02B3 | E0CA        |                  | LDR R,10,12         | 978     |
|      |             | CYCLE            | ANS(.a1,)           | 979     |
|      |             | TEST             | EQU \$              | 980     |
| 02B4 | E0A7        |                  | LDR R,7,10          | 981     |
| 02B5 | E378 2FA0   |                  | LDR DX,8,0+NS,7     | 982     |
|      |             | NEG: DO          |                     | 983     |
|      |             | EQU \$           | I,3,@99             | 984     |
| 02B7 | C103 02BE   |                  | BRC NS(.a1,)=@BP    | 985     |
|      |             |                  | LDR R,8,10          | 986     |
| 02B9 | E0A8        |                  | STR DX,11,0+NS,8    | 987     |
| 02BA | 988B 2FA0   |                  | REPEAT MAINLP       | 988     |
| 02BC | C107 011C   |                  | BRC I,7,@25         | 989     |
|      |             | END              |                     | 990     |
|      |             | EQU \$           |                     | 991     |
|      |             | EQU \$           |                     | 992     |
|      |             | TEST(I=@I+5)>195 |                     | 993     |
|      |             | LDR R,8,10       |                     | 994     |
|      |             | ADD IS,8,5       |                     | 995     |
|      |             | LDR R,10,8       |                     | 996     |
|      |             | CMP I,8,195      |                     | 997     |
|      |             | IFSO: DO         |                     | 998     |
|      |             | EQU \$           |                     | 999     |
|      |             | BRC              |                     | 1000    |
| 02BE | E0A8        |                  | I,6,@103            | 1001    |
| 02BF | 6058        |                  | DUMP DEC(NS,NS+195) | 1002    |
| 02C0 | E08A        |                  | I3,@104             | 1003    |
| 02C1 | B108 00C3   |                  | IS,7,@105           | 1004    |
|      |             |                  |                     | 1005    |
| 02C3 | C106 02CB   |                  |                     | 1006    |
| 02C5 | FE0D 02C8   |                  |                     | 1007    |
| 02C7 | 0017        |                  |                     | 1008    |



| LOC  | OBJECT CODE | CARD IMAGE                                                   |
|------|-------------|--------------------------------------------------------------|
| 02C8 | 2FA0        | @104 EQU \$ Q*NS                                             |
| 02C9 | 30C3        | DC 195+NS                                                    |
| 02CA | BF17        | @105 EQU \$ RETURN                                           |
|      |             | END R,7,1                                                    |
|      |             | END                                                          |
|      |             | @103 EQU \$                                                  |
|      |             | @100 EQU \$                                                  |
|      |             | END                                                          |
| 02C5 | C107 02B4   | BRC 1,7,@95                                                  |
|      |             | END COMPUTE                                                  |
|      |             | @28 EQU \$                                                   |
|      |             | END MAINLP                                                   |
|      |             | @20 EQU \$                                                   |
|      |             | /" ROUTINE BELOW COMPRESSES RESIDUAL PULSES WITHIN THE INPUT |
|      |             | BUFFER. AN ATAC ASSEMBLY LANGUAGE PROGRAM TO PROCESS THIS    |
|      |             | RESIDUAL MAY BE INCORPORATED BETWEEN THE END AND END MAIN    |
|      |             | STATEMENTS UTILIZING THE CODE COLLECTION HEAD. "/            |
|      |             | BP=0 LDR R,11,i2                                             |
| 02CD | E0C6        | ZEROCK: CYCLE                                                |
|      |             | % EQU \$                                                     |
|      |             | @106 TEST @BUF(,@BP+1.)                                      |
|      |             | @108 EQU \$                                                  |
|      |             | LDR R,8,11                                                   |
|      |             | ADD IS,8,1                                                   |
|      |             | LDR DX,2,0+BUF,8                                             |
|      |             | ZERO: DC                                                     |
|      |             | @111 EQU \$                                                  |
|      |             | BRC 1,5,@112                                                 |
|      |             | OP=@BP                                                       |
|      |             | STR D,11,0+DP                                                |
|      |             | LEAVE ZEROCK                                                 |
|      |             | BRC 1,7,@107                                                 |
|      |             | END                                                          |
|      |             | @112 EQU \$                                                  |
|      |             | @109 EQU \$                                                  |
|      |             | BP=@BP+5                                                     |
|      |             | LDR R,2,11                                                   |
|      |             | ADD IS,2,5                                                   |
|      |             | LDR R,11,2                                                   |
|      |             | END ZEROCK                                                   |
|      |             | BRC 1,7,@108                                                 |
|      |             | @107 EQU \$                                                  |
|      |             | CYCLE                                                        |
|      |             | BP=@BP+5                                                     |
|      |             | @113 EQU \$                                                  |
| 02CE | E088        |                                                              |
| 02CF | 6018        |                                                              |
| 02D0 | E382 2000   |                                                              |
| 02D2 | C105 02D8   |                                                              |
| 02D4 | 9C08 31B3   |                                                              |
| 02D6 | C1C7 02DD   |                                                              |
| 02L8 | E0E2        |                                                              |
| 02D9 | 6052        |                                                              |
| 02DA | E028        |                                                              |
| 02DB | C107 02CE   |                                                              |



| LOC  | OBJECT CODE | CARD | IMAGE         | CARDNUM |
|------|-------------|------|---------------|---------|
| 02CD | E0B2        |      | LDR R,2,11    | 1062    |
| 02DE | 6052        |      | ADD IS,2,5    | 1063    |
| 02CF | E02B        |      | LDR R,11,2    | 1064    |
|      |             |      | TEST @BP>3995 | 1065    |
| 02E0 | E0B2        |      | LDR R,2,11    | 1066    |
| 02E1 | B102 0F9B   |      | CMP I,2,3995  | 1067    |
|      |             |      | IFSO: DC      | 1068    |
| 02E3 | C106 02E6   | a116 | EQU \$        | 1069    |
|      |             |      | BRC I,6,a117  | 1070    |
| 02E5 | BF17        |      | RETURN        | 1071    |
|      |             |      | BRC R,7,1     | 1072    |
|      |             |      | END           | 1073    |
|      |             |      |               | 1074    |
|      |             |      |               | 1075    |
|      |             |      |               | 1076    |
|      |             |      |               | 1077    |
|      |             |      |               | 1078    |
|      |             |      |               | 1079    |
|      |             |      |               | 1080    |
|      |             |      |               | 1081    |
|      |             |      |               | 1082    |
|      |             |      |               | 1083    |
|      |             |      |               | 1084    |
|      |             |      |               | 1085    |
|      |             |      |               | 1086    |
|      |             |      |               | 1087    |
|      |             |      |               | 1088    |
|      |             |      |               | 1089    |
|      |             |      |               | 1090    |
|      |             |      |               | 1091    |
|      |             |      |               | 1092    |
|      |             |      |               | 1093    |
|      |             |      |               | 1094    |
|      |             |      |               | 1095    |
|      |             |      |               | 1096    |
|      |             |      |               | 1097    |
|      |             |      |               | 1098    |
|      |             |      |               | 1099    |
|      |             |      |               | 1100    |
|      |             |      |               | 1101    |
|      |             |      |               | 1102    |
|      |             |      |               | 1103    |
|      |             |      |               | 1104    |
|      |             |      |               | 1105    |
|      |             |      |               | 1106    |
|      |             |      |               | 1107    |
|      |             |      |               | 1108    |
|      |             |      |               | 1109    |
|      |             |      |               | 1110    |
|      |             |      |               | 1111    |
|      |             |      |               | 1112    |
|      |             |      |               | 1113    |
|      |             |      |               | 1114    |



| LOC  | OBJECT CODE | CARD IMAGE                                         | CARDNUM |
|------|-------------|----------------------------------------------------|---------|
|      |             | @121 EQU \$                                        | 1115    |
|      |             | @118 EQU \$                                        | 1116    |
| 0308 | C107 02LD   | . END BRC I,7,@113                                 | 1117    |
| 030A | BF17        | .END MAIN BRC R,7,1                                | 1118    |
|      |             | . BRC R,7,1                                        | 1119    |
|      |             | :/***** LINK MODULE FOR EXECUTING PL/ATAC PROGRAM. | 1120    |
|      |             | .\$EXECUTIVE                                       | 1121    |
|      |             | .\$***/                                            | 1122    |
|      |             | MODULE LINK                                        | 1123    |
|      |             | ENTRY LINK                                         | 1124    |
|      |             | DECLARE PROCSTK(64)RAM                             | 1125    |
|      |             | CRG RAM                                            | 1126    |
| 3241 |             | PROCSTK RS = PROCSTK                               | 1127    |
|      |             | CRG ROM                                            | 1128    |
|      |             | LINK                                               | 1129    |
| 0308 | 400C        | EQU \$                                             | 1130    |
| 030C | E100 3241   | LDR IS,12,0                                        | 1131    |
|      |             | LDR I,0,0+PROCSTK                                  | 1132    |
|      |             | CALL MAIN()                                        | 1133    |
| 030E | 601C        | ADD IS,0,1                                         | 1134    |
| 030F | 8DA0        | STK DR,0,11                                        | 1135    |
| 0310 | EDC1 00FS   | I,1,0+MAIN                                         | 1136    |
| 0312 | 8EAC        | BAL DR,0,11                                        | 1137    |
| 0313 | 6FFC        | RET DR,0,11                                        | 1138    |
| 0314 | 5C02        | ADD IS,0,-1                                        | 1139    |
|      |             | LDR RX,2,0,12                                      | 1140    |
| 0315 | BF17        | END LINK R,7,1                                     | 1141    |
|      |             | .EOF EOF EOF EOF EOF EOF                           | 1142    |
|      |             | END                                                | 1143    |
|      |             |                                                    | 1144    |
|      |             |                                                    | 1145    |
|      |             |                                                    | 1146    |



91











## BIBLIOGRAPHY

1. Frazer, P.D., The Electronic Warfare Application of Special Purpose Microprogrammed Minicomputers, M.S.E.E Thesis, Naval Postgraduate School, June, 1974.
2. ATAC, Applied Technology Airborne Computer, V. I,II, Itek Corporation, 1974.
3. Electronic Countermeasures, U.S. Army Signal Corps, 1961.
4. User's Manual, 1st ed., W.R. Church Computer Center, Naval Postgraduate School, Monterey, California, 1970.



INITIAL DISTRIBUTION LIST

|                                                                                                                                                                       | No. Copies |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 1. Defense Documentation Center<br>Cameron Station<br>Alexandria, Virginia 22314                                                                                      | 2          |
| 2. Library, Code 0212<br>Naval Postgraduate School<br>Monterey, California 93940                                                                                      | 2          |
| 3. Department Chairman, Code 57<br>Department of Aeronautics<br>Naval Postgraduate School<br>Monterey, California 93940                                               | 2          |
| 4. Professor Stephen Jauregui, Jr., Code 52Ja<br>Department of Electrical Engineering<br>Naval Postgraduate School<br>Monterey, California 93940                      | 10         |
| 5. Commander Navy Security Group Command (G-80)<br>Navay Security Group Headquarters<br>3801 Nebraska Avenue, N.W.<br>Washington, D.C. 20390<br>ATTN: CDR H. Orejuela | 2          |
| 6. Commander<br>Naval Electronics Systems Command (PME 107)<br>Naval Electronics Systems Command Headquarters<br>Washington, D.C. 20360<br>ATTN: LCDR A. Sagarian     | 2          |
| 7. Commander<br>Naval Electronics Systems Command (PME 107)<br>Naval Electronics Systems Command Headquarters<br>Washington, D.C. 20360<br>ATTN: LCDR R. Shields      | 1          |
| 8. Commander<br>Naval Air Systems Command<br>Naval Air Systems Command Headquarters<br>Washington, D.C. 20361<br>ATTN: Air-53321A<br>CDR V. McCauley                  | 1          |



9. Applied Technology 1  
645 Almanor Avenue  
Sunnyvale, California 94086  
ATTN: Mr. J. Houston
10. LT Robert D Stanga, Jr. 1  
1105 Sparrow Drive  
Pt. Mugu, California 93042

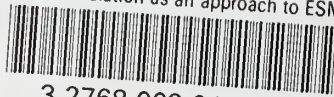


Thesis  
S6736 Stanga  
c.1 Digital simulation  
as a approach to ESM  
software development.  
16 MAY 76  
10 AUG 79  
12 OCT 89  
163613  
S12298  
14194

Thesis  
S6736 Stanga  
c.1 Digital simulation  
as a approach to ESM  
software development.  
163613

thesS6736

Digital simulation as an approach to ESM



3 2768 002 01623 0

DUDLEY KNOX LIBRARY